UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF TEXAS
DALLAS DIVISION

SBC TECHNOLOGY RESOURCES, INC.,)
)
Plaintiff,) HONORABLE David C. Godbey
)
vs.) CIVIL ACTION NO. 303-CV-418-N
)
INRANGE TECHNOLOGIES CORP.,)
ECLIPSYS CORP.; and)
RESOURCE BANCSHARES)
MORTGAGE GROUP, INC.,)
)
Defendants.)
)

INRANGE TECHNOLOGIES CORPORATION'S CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

TABLE OF LEGAL AUTHORITIES

Case Law

Statutes

I. INTRODUCTION

A. The Scope of This Case

There is at issue in this case, one patent and one accused product. The scope of this case is encapsulated in the title of U.S. Pat. No. 5,530,845 ("the '845 patent"): "Storage Control Subsystem Implemented With An Application Program On a Computer." See Ex. A, the '845 patent. In the Abstract, the inventors stated:

> A storage controller is disclosed which may emulate several types of specialized host specific and/or storage device specific storage controllers. The storage controlling system can transfer information between one or more different types of target units and one or more channels of at least one host. The system is provided with a computer, which includes a first interface, a second interface, and a programmable storage controller . . . .

Id. (emphasis added).

As the title and abstract suggest, the '845 patent's claims relate to only one subpart that may play a part in a broader storage area network (SAN), namely: the storage control subsystem for remote data storage and retrieval, and a particular way of implementing that subsystem using a particular "programmable storage controller."

Contrary to insinuations in SBC's Opening Claim Construction Brief ("SBC's Brief"), this case is not broadly about SANs, but is instead limited to a particular type of storage control subsystem that may be a part of a SAN. While it is important to understand the concept of a SAN in relation to the '845 patent, SBC's attempt to stretch that patent – and thereby tangentially engulf Inrange's products used in SANs – is improper.

The only Inrange product accused by SBC of being a programmable storage controller is Inrange's FC/9000 Fibre Channel switch. [1] The FC/9000 provides a function within a SAN that

[1] Sometimes referred to as a "fabric switch" or "director" in the context of a SAN, products like the FC/9000 "enable any storage device to be connected to any computing device for the duration of a data transfer operation." See Thomas C. Jepsen, Distributed Storage Networks 10 (John Wiley & Sons, Ltd. 2003) (hereinafter "DSN")

is distinct from the function of a "programmable storage controller" – a term common to ***every*** claim in the patent.

Using Figure 4 from the '845 patent, one can see the logical relationship of the claimed programmable storage controller (item 14) to other elements (host computer (items 28, 30) and storage facilities (items 16, 32)) commonly used in a SAN.

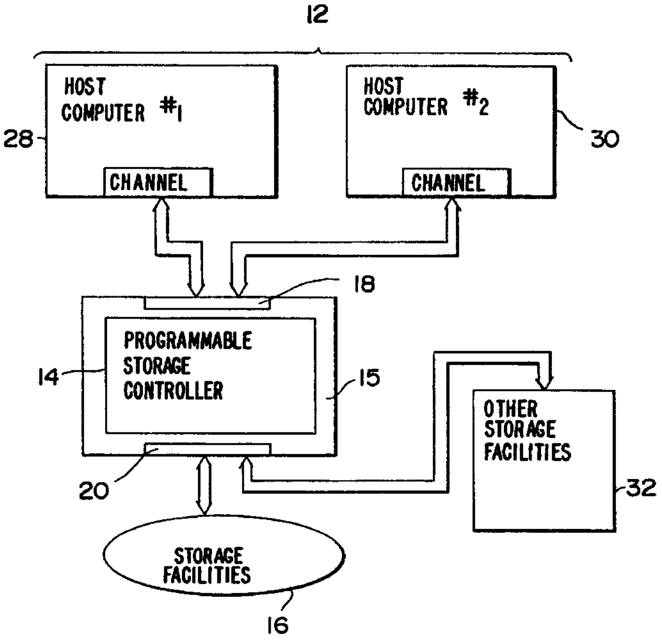


FIG. 4

As shown in the above figure, the FC/9000 product would lie, in logical relationship, in the channel between the host computer (28) and the first interface (18) for the claimed programmable storage controller (14).

The dispute here boils down to this: SBC seeks to have the claims of this "programmable storage controller" patent construed to cover the Inrange FC/9000 Fibre Channel switch. SBC's

(Excerpts attached at Ex. B). "A storage director is a specialized type of fabric switch which provide enhanced management and reliability features such as duplicated switch fabrics and power supplies." Id.

attempt to broaden the claims is unsupported by the language of the claims, the patent specification, and the file history.[2]

What is the claimed "programmable storage controller" for a host computer system which the inventors expressly claim as their invention? (Ex. A, 1:11-14).[3] SBC has basically punted by offering no proposed construction. SBC has presumably left it to Inrange to assist the Court in defining and construing this, and other, important claim language of claims 1 and 34 of the '845 patent.

B. General Storage Controller Technology

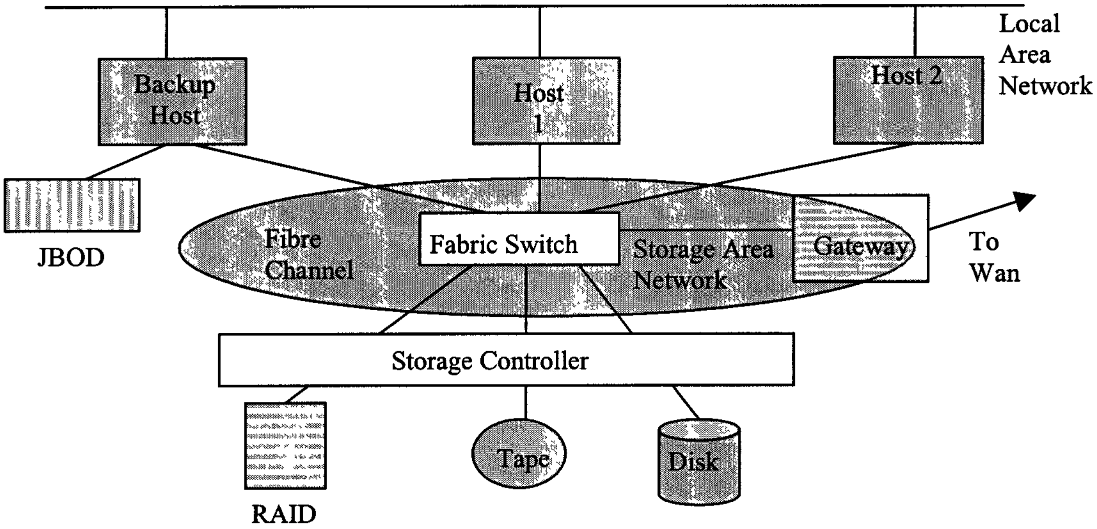
To understand the context in which the '845 patent was granted, it is useful to have a basic knowledge of the working components of a SAN.

As electronic data storage needs of a business organization increase, it is desirable to have the business's central computers freed up from massive data storage requirements. External or remote data storage allows for both lower cost storage of electronic data and scalability of external or remote storage facilities. Ex A, 1:46-49; see also 13:53-60. Thus, there has developed over time, especially by IBM and other large mainframe computer manufacturers, a series of products that together comprise a SAN. As defined in one industry dictionary, a SAN is "a network whose primary purpose is the transfer of data between computer systems and storage elements and among storage elements. . . . A storage system consisting of storage elements, storage devices, computer systems, and/or appliances, plus all control software communicating over a network." Ex. C, A Dictionary of Online Storage Networking Terminology, Storage

[2] Inrange is not seeking to define the asserted claims only in connection to the accused device, but SBC has summarily told the court that the accused device is a storage controller (SBC's Brief, at 1), and that conclusory accusation deserves response and correction.

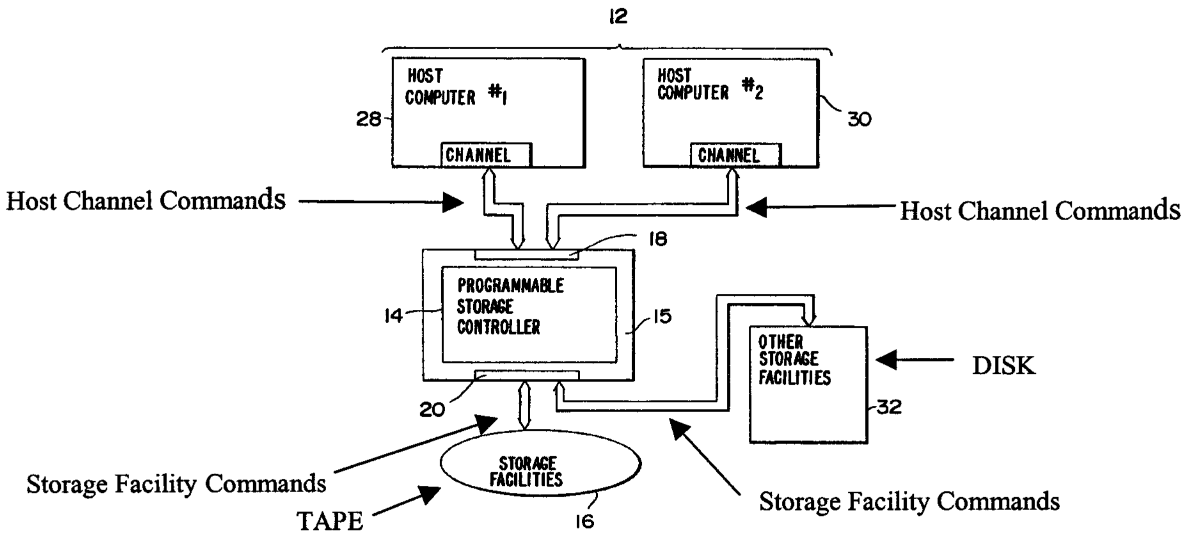
[3] For purposes of this brief, the '845 patent's column and line numbers are cited in that order and separated by a colon (XX:XX).

Networking Industry Association (SNIA) at http://www.snia.org/education/dictionary (last visited Oct. 25, 2004). A diagram depicting elements that can be included in a SAN shows the logical connection between various host computers and storage elements:



In this SAN environment, host computers generally use remote (or peripheral) storage facilities on which to store data. These storage facilities can be a single storage device (such as a disk or tape drive), or a combination of storage devices. As explained in the ‘845 patent, storage controllers act as an intermediary for commands to and control of data between the host computer and the storage facility. Ex. A, 1:19-21.

The following diagram shows the traditional relationship between these elements in a SAN:



Mainframe Storage Area Network (Modified FIG. 4 From the '845 Patent)

As the '845 patent makes clear, the role of the claimed programmable storage controllers in this environment is to assume control over the data and commands to ensure that data reaching the storage controller is successfully stored on the targeted storage facility and successfully retrieved at a later date. Id. at 7:30-39. Together, the programmable storage controller and the storage facility form a storage subsystem. Id. at 9:38-42.

One of the primary roles played by storage controllers is to translate the channel language commands sent by the host into the commands and addressing scheme used by the storage facility, thereby sheltering the host computer from the operational complexity of the storage facility. R. Spaulding, Storage Networks: The Complete Reference, 83 (2003) (hereinafter "TCR"). (Excerpts attached at Ex. D). This so-called translation step (common to all claims in the '845 patent) is necessary because the invention claimed is intended to support communication and execute commands sent by a particular host computer but not typically understandable to the storage facility(ies) where the data is maintained. See Ex. A, 2:22-29; 4:55-58.

As described in the ‘845 patent, a host computer (items 28 and 30), uses a host adapter to send input/output (I/O) commands for delivery and action upon data to storage facilities over a communication pathway known as a channel. Ex. A, 1:22-8. These I/O commands are in a format appropriate for the particular channel’s language or protocol. Id. 14:36-40. In the claimed invention, the I/O commands, sent in a particular command language, are not understood directly by the storage devices; therefore, they must be interpreted by a programmable storage controller. Id. at 1:28-30.

For example, the channel commands are sent from the host along a channel to the programmable storage controller. The programmable storage controller interprets, translates, and implements the channel command by sending to the storage facility one or more data address and request instructions that can be understood by the storage facility. The storage facility then executes the instruction (e.g. read data from disk/tape, write data to disk/tape) and returns a status response back to the programmable storage controller. The storage controller translates the storage facility’s response into a channel command that would be understood by the host, and then transmits that channel command response to the host. This same logical transfer of I/O data prevails whether the host computer and storage facility(ies) are six feet away from each other or whether (in modern fibre channel SANs) the host(s) and storage facility(ies) are several miles apart. See DSN at 4-5 (discussion of history of storage facilities) (Ex. B).

A storage controller may control only one storage facility or it may control a multiple storage facilities (as depicted in FIG. 4 of the ’845 patent). For example, a storage controller may control an array of disks such as a RAID[4] storage array. See TCR at 228-9 (Ex. D). A RAID storage controller shields the host computer from the complexity of the RAID array by

[4] RAID is a common term for Redundant Array of Inexpensive (or Independent) Disks. A Dictionary of Online Storage Networking Terminology at http://www.snia.org/education/dictionary. Ex. E.

presenting itself to the host computer virtually as a single device instead of the actual array of storage devices. The RAID storage controller takes ownership of the data, interprets the channel command message sent by the host computer, and converts from the channel commands received from the host into the instructions that are necessary to actually save and retrieve the data from the array of disks storage devices. See TCR at 228-9 (Ex. D); T. Clark, Designing Storage Area Networks, 112-115 (2d. ed. 2003) (hereinafter "DSAN") (Ex. F).

In a SAN environment, different types of media (including Extended Count Key Data (ECKD) disks or fixed block disks that are randomly accessible and tape-sequentially accessible) may be employed. Ex. A, 9:12-27. Tape and disk each employ different channel protocols and operations. Id. DSAN at 410, 509 (Ex. F). Without a "programmable storage controller," each type of target unit media would need to employ its own specialized dedicated storage controller. Ex. A, 1:52-2:1.

C. What is the Invention?

In the inventor's own words:

> The present invention relates to a storage controller for a host computer system. More particularly, the present invention relates to a programmable storage controller implemented on a computer which is capable of emulating one or more storage device specific and/or host specific storage controllers.

Ex. A, 1:10-15. As reflected by the plain language of the patent specification and the patent prosecution history discussed infra, the inventors did not claim to invent the storage controller (or even the programmable ones). Ex. A, 1:65 – 2:14. The '845 patent discloses instead an improvement to storage controllers that can emulate multiple types (e.g. disk and tape) of storage controllers in a particular manner. Id.[5]

[5] Different types of storage media are described in the patent as ECKD disks, fixed block disks, and tape (sequential access) storage devices. Ex. A, 9:12-27. A disk device is fundamentally different from a tape device in that a disk

As will be described in further detail, the ability to emulate multiple types of target unit storage controllers by using an application program on a computer did not alone distinguish the '845 patent over the prior art. Instead of the broader claims initially sought, the applicants had to make changes and concessions regarding their originally-filed claims. What resulted as the issued claims of the '845 patent was a specific set of attributes of the "programmable storage controller" claimed, which the court is now asked to construe. See Markman v. Westview Instruments, Inc., 517 U.S. 370 (1996).

D. The Application For the '845 Patent

The '845 patent is based upon United States patent application serial no. 08/373,896, which was filed January 17, 1995, which in turn was a continuation for United States patent application serial no. 07/882,010, first filed on May 13, 1992, but later abandoned. Ex. A. The broadest claim in the first (parent) application in 1992 attempted to cover any "programmable" storage controller that contained a first channel interface and a second interface connected to one or more target units. Ex. G, I001517.[6] In a first office action, the examiner rejected all of the claims. Id. at I001454-63. Specifically, the broad pending claim 1 was rejected as anticipated by U.S. Patent No. 4,803,623, which was issued to Klashka et al. on February 7, 1989 ("the Klashka patent") (Ex. H). Ex. G at I001456.

The Klashka patent contained a description of the then-existing prior art and provided diagram of a storage control subsystem embodiment of the Klashka invention. See Ex. H and FIGS. 1 and 2 thereof. The examiner described Klashka as a prior art reference that "discloses

drive is randomly addressable while a tape drive is written to and read from sequentially. Id.; DSAN at 410, 509 (Ex. F). Therefore, the channel commands used to communicate with a disk device are different from those used to communicate with a tape device. Similarly, ECKD disk drives format data on the drives fundamentally differently than fixed block drives. Each type of disk therefore requires different channel commands. Ex. A, 9:18-27.

[6] The entire prosecution history of the '845 patent is submitted herewith as Ex. G, with internal page references to the corresponding production control number (e.g. "I001285").

the invention as claimed, including: a first interface . . ., a second interface . . ., a programmable storage controller . . ." Ex. G, I001456 (emphasis added). The forty pending claims were rejected by the examiner, in light of both anticipation by Klashka and in light of other references as well. Id. at I001454-61.

In a January 21, 1994, interview with the examiner, the applicants responded by defining their invention as limited to using a general-purpose computer rather than a stand-alone microprocessor. Id. at I001430. The applicants also responded to the First Office Action with an Amendment on May 12, 1994, purporting to address the prior art rejections. Id. at I001432-46. The applicants canceled the broad independent claims and replaced them with new independent claims. The applicants also argued in response to the office action by seeking to distinguish Klashka (which they argued taught a storage control system implemented with specific hardware architectures) from their new claims. The applicants argued that the new claims require, and the prior art does not show, "the use of a general purpose computer to implement a storage controller, providing operation and application systems on the general purpose computer, and implementing the programmable storage controller with an application program within the application system." Id. at I001442. (emphasis added).

Despite the interview and despite the applicants' May 12, 1994, claim amendment and office action response, the examiner again rejected all of the pending claims with a final office action dated July 14, 1994. Id. at I001421-28. The examiner stated the applicants' argument about Klashka was not persuasive. Id. In response to the final office action, the applicants filed yet another amendment on October 14, 1994, and again argued that an application program on a general-purpose computer is different from the loadable device drivers of Klashka. Id. at I001403-19. The examiner responded with an Advisory Action stating again that the applicants' arguments did not overcome the rejection. Id. at I001401. The applicants then filed a Notice of Appeal on November 14, 1994. Id. at I001396-7.

The applicants followed that Notice of Appeal with a January 17, 1995, File Wrapper Continuation Application – and they abandoned the earlier application. Id. at I001393-4. The applicants also included a Preliminary Amendment in their Continuation Application that canceled the independent claims and added yet another set of new claims, including four independent claims. Id. at I001363-71. These new independent claims added a claim element requiring the translation of "channel programs and commands from a channel specific format to a generic format of said programmable storage controller that includes generic address and request information." (emphasis added). The applicants argued for patentability by stating the prior art, including the Klashka patent:

> fails to disclose a programmable storage controller that emulates a plurality of types of target unit specific storage controllers, wherein the programmable storage controller is implemented with an application program and a computer, and comprises a plurality of controller emulators that comprise means for translating the channel programs and commands from a channel specific format to a generic format of the programmable storage controller.
>
> Id. at I001370 (emphasis added).

The PTO responded with a Notice of Allowance (Id. at I001353) and a Statement of Reasons for Allowance dated November 13, 1995. (Id. at I001354-55). The examiner's statements said:

> prior arts of record do not teach a programmable storage controller that emulates a plurality of types of target unit specific storage controllers. This programmable storage controller is implemented with application programs and a general-purpose computer that allows for the user to switch between these application programs without reloading or changing the operating system, or physically modifying the hardware configuration. This programmable storage controller, further, comprises a plurality of controller emulators for translating channel programs and commands from a channel specific format [address and requested information] to a generic format.

> The examiner considers the applicants' claims . . . to be allowable based on the claim interpretation and the aforesaid prior arts of record.

Id. (emphasis added, bracketed text in original).

The applicants filed Comments on Examiner's Statement of Reasons for Allowance, dated February 15, 1996, in which they stated no disagreement with the examiner's reasons for allowance and merely restated, for clarity, that: "the claims in the present application recite a combination of features, and that patentability of these claims is *also* based on the totality of the features recited therein, which define over the prior art of record. Id. at I001336-7 (emphasis added).

II. LEGAL STANDARDS FOR CLAIM CONSTRUCTION

A. Inrange's Requested Constructions

Inrange asks the court to construe only the claim terms that are in dispute, nothing more and nothing less. Indeed, SBC recognizes that the court needs to construe those terms placed in dispute. See SBC's Brief at 9. Nevertheless, rather than provide the court with a true section on "The Law of Claim Construction" as SBC's section title suggests, SBC presents a straw man argument that Inrange is somehow attempting to avoid the meaning of the claims. By continually parroting back the claim language as its "construction," SBC erroneously suggests the disputed claims should not be interpreted by the Court but rather later divined by the jury. In short, SBC has decided to prejudge and dismiss any construction of its claims as a "rewrite gambit." Id. In doing so, SBC seeks to avoid its obligation to offer any interpretation and further belies its intention to avoid the proper limitation of its claims through obfuscation of the claim construction process.

As a practical matter, SBC is also incorrect in asserting Inrange is somehow seeking construction of nearly every word in the claims. As SBC's Brief recognizes, it is necessary to define several claim passages in context, and in doing so, many terms internal to those passages are also necessarily defined. See e.g., SBC's Brief defining "**translating** said **channel programs** and commands from a **channel specific format** to a **generic format** of said **programmable storage controller**" at 20-21. Thus, many of the individual terms for which Inrange seeks construction actually form a nested set of constructions for the broader, critical elements of the asserted claims. Inrange is neither seeking construction of any term not in dispute nor seeking to construe every term in the asserted patent claims.[7]

B. General Claim Construction Law[8]

Contrary to SBC's pronouncements (SBC's Brief, at 14), precedent from the Federal Circuit over the last decade has made clear that during claim construction a court cannot simply ignore a patent's specification and file history. "It is well-settled that, in interpreting an asserted claim, the court should look first to the intrinsic evidence of record, *i.e.* the patent itself, including the claims, the specification and, if in evidence, the prosecution history." Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582 (Fed. Cir. 1996). "Such intrinsic evidence is the

[7] Inrange also notes that many of the specific individual terms listed in Exhibits B and C to SBC's Brief were terms the former co-defendant, Eclipsys Corp., had considered in need of construction for purposes of its own defense. SBC settled its claims against Eclipsys after the filing of its brief; therefore, much of SBC's rhetoric regarding the number of terms in dispute is rendered moot and inapplicable to Inrange.

[8] This is not an opportune time to be construing litigated patents. The Federal Circuit recently took en banc consideration of several important tenets of claim construction on issues that have been percolating within the Federal Circuit and the district courts over the past several years. See Phillips v. AWH Corp., 376 F.3d 1382 (Fed. Cir. 2004) (courtesy copy attached as Ex. S). The continuing place and use of dictionaries in the claim construction process; the interpretation, or not, of "ordinary meaning" terms; and whether any deference should be afforded to the district court's claim construction are among the topics for the Federal Circuit's full consideration. Id. Many amicus curiae briefs have been submitted. See e.g. Brief For the United States as Amicus Curiae and Brief for Amici Curiae Intel Corp., IBM Corp., Google, Inc. Micron Tech., Inc., and Microsoft Corp. available at http://www.patentlyobviousblog.com/2004/09/phillips_.html (Sept. 23, 2004). In the absence of a ruling in the Phillips case any time soon, Inrange will proceed under the existing authorities and employ current controlling claim construction precepts from the Federal Circuit.

most significant source of the legally operative meaning of disputed claim language." Id. A claim term is given its ordinary and customary meaning, if any, as understood by one of ordinary skill in the art, unless the patentee has chosen to use the term in some other manner. Allen Eng'g Corp. v. Bartell Indus., Inc., 299 F.3d 1336, 1344 (Fed. Cir. 2002).

A court must look to the specification to determine whether the patentee has been his own "lexicographer" and ascribed a particular definition to a claim term or has otherwise assigned any special meaning to a claim term. Vitronics Corp., 90 F.3d at 1582; Bell Atlantic Network Servs., Inc. v. Covad Communications Group, Inc., 262 F.3d 1258, 1268-1273 (Fed. Cir. 2001). Even if a claim term is not given a special meaning in the specification, if "the claim language is not clear on its face, then our consideration of the rest of the intrinsic evidence is directed to resolving, if possible, the lack of clarity." Interactive Gift Express, Inc. v. Compuserve Inc., 256 F.3d 1323, 1331 (Fed. Cir. 2001). Further, "Where the specification makes clear that the invention does not include a particular feature, that feature is deemed to be outside the reach of the claims of the patent, even though the language of the claims, read without reference to the specification, might be considered broad enough to encompass the feature in question." SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc., 242 F.3d 1337, 1341 (Fed. Cir. 2001).

In addition to the specification, the court must look to the patent's entire file history to determine whether the patentee has given a claim term special meaning or disclaimed the scope of any claim. The prosecution history "is often of critical significance in determining the meaning of the claims" because it "contains the complete record of all the proceedings before the Patent and Trademark Office, including any express representations made by the applicant regarding the scope of the claims." Vitronics Corp., 90 F.3d at 1582. In construing the claims, the Court must "exclude any interpretation that was disclaimed during prosecution." Southwall Tech., Inc. v. Cardinal IG Co., 54 F.3d 1570, 1576 (Fed. Cir. 1995); Elkay Mfg. Co. v. Ebco Mfg. Co., 192 F.3d 973, 980 (Fed Cir. 1999) (finding a claim term limited based upon the patentee's arguments to overcome prior art *and* the examiner's comments); Bell Atlantic Network Servs., Inc., 262 F.3d at 1268. "Claims may not be construed one way in order to obtain their allowance and in a different way against accused infringers." Southwall Tech., Inc., 54 F.3d at 1576; Spectrum Int'l, Inc. v. Sterilite Corp., 164 F.3d 1372, 1378-1379 (Fed. Cir. 1998).

Instead of looking to the intrinsic, contextual evidence, SBC erroneously relies heavily on dictionary definitions. SBC's heavy reliance on these dictionaries is improper for two reasons.

First, even the leading Federal Circuit case promoting the use of dictionaries recognizes that "Indeed, the intrinsic record may show that the specification uses the words in a manner clearly inconsistent with the ordinary meaning reflected, for example, in a dictionary definition. In such a case, the inconsistent dictionary definition must be rejected." Texas Digital Systems, Inc. v. Telegenix, Inc., 308 F.3d 1193, 1203 (Fed. Cir. 2002). Even where the intrinsic evidence gives no special meaning to a term, "the intrinsic record must always be consulted to identify which of the different possible dictionary meanings of the claim terms in issue is most consistent with the use of the words by the inventor." Texas Digital Systems, Inc., 308 F.3d at 1203.[9]

Second, a heavy reliance on dictionary definitions rather than the specification to define the ordinary meaning of terms is unsound given the Federal Circuit's recent decision to review

[9] By analogy, looking to the first definition of "patent" as a noun in Merriam-Webster, one finds "an official document conferring a right or privilege." (See Ex. I) Under that definition, the '845 patent, a driver's license, a passport, and a building permit could all be considered "patents." As discussed elsewhere, the court should avoid, where possible, in this context any reliance on general purpose dictionaries for construing "programmable storage controller" or other terms. Texas Digital Systems, Inc., 308 F.3d at 1203.

the role dictionaries play in claim construction. Given the Federal Circuit's order to brief questions directly related to the deference the specification should be given over dictionaries in determining a claim's ordinary meaning, claim construction in this case is best performed with a focus on the intrinsic evidence rather than a reliance on dictionaries.[10]

C. Claim Limitations From the Prosecution History of Related Patents

On claim construction, federal courts always consult the prosecution history to confirm the construction of terms. Vitronics Corp., 90 F.3d at 1582. The entire record is relevant for this purpose. Thus, where a claim element is in both the parent (first filed) and the child (later filed) patent application, limitations from the parent's file history can be imported into the child patent's claims. Biovail Corp. International v. Andrx Pharmaceuticals, Inc., 57 U.S.P.Q.2d 1813, 1816 (Fed. Cir. 2001) (the prosecution history of the parent application applies to the child patent's claim construction where "admixture" limitation was added to the parent application as an amendment after final rejection and "admixture" limitation appears in a similar context in the child patent); Wang Labs. v. America Online, Inc., 197 F.3d 1377, 1384 (Fed. Cir. 1999) (statements made to distinguish prior art in a parent application apply to the child application where the argument concerning the prior art applies to the common subject matter between the patents); Jonsson v. Stanley Works, 903 F.3d 812, 818 (Fed. Cir. 1990) (same).

Additionally, a claim term may be limited in construction where, after receiving a rejection, the patentee simply filed a new continuation-in-part application and narrowed the scope of the claims to avoid the prior art. Mark I Marketing Corp. v. R.R. Donnelley & Sons Co., 66 F.3d 285, 291-292 (Fed. Cir. 1995) (finding the patentee had surrendered the patent scope that it attempted to avoid through filing new patents rather than responding to the office

[10] See Fn. 8, supra.

actions); Desper Products, Inc. v. Qsound Labs, Inc., 157 F.3d 1325, 1333-1335; 1338-1339 (Fed. Cir. 1998) ("[P]rosecution history estoppel cannot be avoided by filing a continuing application with narrowed claims rather than responding directly to an outstanding rejection.").

D. Construction of Means-Plus-Function Elements Under 35 U.S.C. § 112¶6

There are, the parties agree, certain asserted claims with means-plus-function elements. As a matter of law, means-plus-function elements are limited to the corresponding structure disclosed in the specification and equivalents thereto. Valmont Indus., Inc. v. Reinke Mfg. Co., 983 F.2d 1039, 1042 (Fed. Cir. 1993). When a claim limitation recites the word 'means,' § 112, ¶ 6 presumptively applies. Kemco Sales, Inc. v. Control Papers Co., 208 F.3d 1352, 1361 (Fed. Cir. 2000). If an element is in means-plus-function form, the Court next identifies the function explicitly recited in the claim. Cardiac Pacemakers, Inc. v. St. Jude Med., Inc., 296 F.3d 1106, 1113 (Fed. Cir. 2002). The Court then determines what structure, if any, is disclosed in the specification that corresponds to the claimed function. Id.; B. Braun Med., Inc. v. Abbott Labs., 124 F.3d 1419, 1424 (Fed. Cir. 1997). "Structure disclosed in the specification is 'corresponding' structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim." Id.; Fonar Corp v. General Elec. Co., 107 F.3d 1543, 1551-1552 (Fed. Cir. 1997) (explaining that although the specification states generally that other wave forms may be used, it fails to specifically identify those wave forms; § 112, ¶6 claim is thus limited to the gradient wave from actually disclosed).

III. ARGUMENT

Only independent claims 1 and 34 are asserted against Inrange's FC/9000 Fibre Channel switch products. (See SBC's Brief at 2).[11] Key elements of Claims 1 and 34 of the '845 patent are substantially similar in most respects even though claim 1 is an apparatus claim and claim 34 is a method claim. Consequently, unless otherwise noted, Inrange proposes the following claim constructions for elements in both claims.

A. Programmable Storage Controller

SBC artfully dodges around the disputed claim term that is at the essence of what it purports to have invented: a particular type of "programmable storage controller." SBC acknowledges that this term is on the parties' list of disputed terms, but then treats it as though "programmable storage controller" need not be construed (or even understood to construe other disputed terms). SBC's Brief at 12-13. SBC assumes the definition of what a storage controller is and glosses over any explanation at all of how to construe this claim element in the context of the overall patent. SBC proceeds instead directly to discussion of the "emulate" limitation. Id. SBC's tactic is untenable.

By avoiding any definition of what a storage controller is, SBC hopes to point vaguely to the accused product, Inrange's FC/9000 and simply denominate it as a "programmable" storage controller. See SBC's Brief at 1. As Inrange will show, the FC/9000 is what is known in SAN architecture as a switch; it is not a programmable storage controller. Because SBC bears the ultimate burden of proof on infringement, it will have to prove that the FC/9000 has the required attributes of a "programmable storage controller," together with all other claim limitations of the

[11] SBC's Brief attempts to draw a distinction between the asserted and non-asserted claims of the '845 patent. See SBC's Brief at 14. From the context of the statement, SBC appears to have limited its infringement charge to claims 1 and 34 and is not asserting independent claims 45 and 47 against Inrange.

properly-construed '845 patent. Simply calling something a "programmable storage controller" does not make it so. Although claim construction should not be done by specific reference to the accused product, the court should be made aware of the FC/9000's actual role in a fiber channel SAN. The FC/9000 is a director-class switch.[12] See product brochure attached to SBC's Brief. SBC wants to call the FC/9000 a "programmable storage controller," but SBC cannot now escape the consequences of avoiding any attempt to define "storage controller" (or for that matter "programmable storage controller"). The term has a central, disputed meaning in the context of SBC's specification and patent infringement allegations made in this case. Accordingly, the court should construe that term.

In the allowed claims of the '845 patent, the required programmable storage controller uses a computer under the control of an application program, with the application program operating multiple storage controller "emulators." Ex A, 9:12-58. Each of these emulators handles the communication with and control of a particular type of storage facility (e.g., an ECKD disk, fixed block disk, or tape device). Id. The '845 patent's programmable storage controller is primarily an application program, employed in an operating system, that runs one or more storage controller emulators, with each storage control emulator in turn being able to emulate the functionality of one type of target unit storage controller. Id. 7:57-58; 9:6-12. Like other previously-known storage controllers, the '845 patent's programmable storage controller controls the storage devices in accordance with addresses and commands issued by the host computer (Id. 8: 35-48), takes ownership of the data and responsibility for ensuring that data is successfully stored and retrievable (Id. 8:38-43), and translates the channel commands received

[12] In this context, a switch is defined as: "A network infrastructure component to which multiple nodes attach. Unlike [a hub], switches typically have internal bandwidth that is a multiple of link bandwidth and the ability to rapidly switch node connections from one to another." SNIA Dictionary, Ex. J.

from the host computer into the instructions and addressing scheme understood by the storage facility (Id. 9:55-62), thereby shielding the host computer from the operational complexity of the storage device. Ex. G, I001406. ("These benefits of the claimed invention are achieved by the features recited in the claims, including . . . an application program, where the application system has an application program which controls exchanges of storage data to and from the target units.") (emphasis added).

The programmable storage controller "interprets the commands and manipulates the storage facilities to satisfy a request." Ex. A, 1:28-30. "In addition, a set of equipment controllers is provided [within the programmable storage controller] which interpret channel program commands and status information and which control data transfer to and from the storage facilities in accordance with the channel program command." Id. at 3:20-24. The addressing scheme of a host system (assigning each block of data a device address) creates "address parameters" that "are then translated by a disk controller into the actual physical address based on the controller's knowledge of the peripherals attached to it. The host operating system must receive consistent responses/results in accordance with the request it makes." Id. at 7:30-37.

Thus, the claimed programmable storage controller acquires "ownership" of the data it receives (together with the corresponding channel specific address and command) from the host computer, and it is the storage controller that is responsible for ensuring that data is successfully stored on, and later able to be retrieved from, the target unit storage facility. For example, as explained by the inventors in the '845 patent, among the many functions (such as compression/decompression (see Ex. A, 3:63-4:3) and intercontroller services (see Ex. A, 2:30-35 and FIG. 2)) that can be incorporated, the claimed programmable storage controller takes responsibility for maintaining a "read" cache where operations can sometimes be fulfilled by reading the requested data directly from the cache, thus eliminating the time it would take to read the data off of the storage facility itself. Ex. A, 12:13-35. In this example, part of the programmable storage controller's "ownership" of the data is taking responsibility for ensuring that the cache accurately reflects the data stored on the storage facility. Id. 12:32-49.

Therefore, Inrange contends that "programmable storage controller" should be defined as follows:

> The logical control system, implemented in programmable computer software and corresponding configured hardware, that receives channel commands from a host computer and ensures the specified delivery, storage, and retrieval actions, by translating those channel commands of the host into the instructions and addressing scheme understood and carried out by the targeted storage facility(ies) under its control, thereby shielding the host computer from the operational complexity of the targeted storage facility(ies).

It would be wrong , and SBC has not suggested doing so, to look to any non-technical dictionaries for guidance on the construction of "programmable storage controller" because of the highly technical nature of the SAN architecture, the established art over which the claims were distinguished, and the specific functionality of the programmable storage controller claimed in the '845 patent. Vitronics Corp., 90 F.3d at 1582; Texas Digital Systems, Inc., 308 F.3d at 1203.

For its part, SBC has only discussed the "emulation" aspect of the claimed programmable storage controller. SBC's Brief at 13-14. Relying on one technical dictionary, SBC suggests that "emulate" means "to represent . . . by a model that accepts the same inputs and outputs as the system represented." Id. at 13. While this definition of "emulate" is not inconsistent, (representing a different system by accepting the same inputs and outputs), it is incomplete. The emulation here is of "a plurality of target unit specific storage controllers," so the emulation is context-specific: the system(s) represented are target unit specific storage controllers, so the "programmable storage controller" must control data (whether on disk or tape) in exactly the same manner as if it were such a target unit specific controller. See Ex. A, 2:26-29.

B. Translation to a Generic Format

The programmable storage controller of claim 1 in SBC's '845 patent includes a plurality of controller emulators "comprising means for translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller that includes generic address and request information." (emphasis added).[13] Similarly, claim 34 contains a method step for controlling data exchanges "comprising translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller, said generic format including a generic address and request information." Therefore, both claims contain a means for, or a method of:

- translating channel programs and commands
- from a channel specific format
- to a generic format of the programmable storage controller
- where the generic format includes both generic address and generic request information

This "translation to a generic format" limitation was not part of SBC's original claims. It was added after the examiner said it was required for patentability: to overcome anticipatory prior art storage controllers. According to the examiner, there was nothing new in SBC's claimed storage control subsystem, until, among other things, this "generic translation" language

[13] This overall element is written in means-plus-function language and is given further definition in the brief.

was added to the pending claims. Ex. G, I001353-55. This important difference is one key reason SBC holds a patent, instead of just an already well-known storage controller concept.[14]

To understand this claim element, the terms "translating," "channel specific format," and "generic format" should be defined. SBC instead presents a circular argument whereby the words of the claims are merely repeated as its definition.[15] In doing so, SBC makes two mistakes: 1) it suggests claims terms that may be given their ordinary meaning need no construction, rather than a construction consistent with the ordinary meanings; and 2) it asserts that each of the terms that form the definition of this element themselves have a clear ordinary meaning. Inrange's claim construction of this claim element provides a definition, rather than a repetition, of the claim language.

1) "Translating"

A proper construction of "translating" is found in the ordinary meaning of the term as defined by technical dictionaries and the '845 patent specification. Computer dictionaries define translating as follows:

Source	Definition
Computer Desktop Encyclopedia	To change from one language into another; for example, assemblers, compilers and interpreters translate source language into machine language (Ex. K)
Microsoft Computer Dictionary	In programming, to convert a program from one language to another. Translation is performed by special programs such as compilers, assemblers, and interpreters (Ex. L)

[14] For purposes of this brief, Inrange makes no arguments, admissions, or even comments regarding the '845 patent's invalidity. Inrange instead recounts these facts only to show that the prosecution history is important for claim construction.

[15] This Court's scheduling order specifically set SBC's Brief as due first, followed by a response by Inrange and a reply by SBC. SBC's choice (or strategy) to throw up its hands and offer little if any claim construction on disputed claims effectively would reverse the briefing order so that SBC will have the final opportunity to offer its true claim construction arguments for the first time in its reply with no opportunity for rebuttal by Inrange. If SBC chooses to use its reply brief to sandbag Inrange by offering new or different interpretations for the elements in dispute, Inrange respectfully requests the opportunity to rebut these new arguments with a succinct surreply and oral arguments on the claim construction issues.

Each definition includes changing some object, either represented data or a programmed command, from one prescribed language to another. This is consistent with the express language of the '845 patent specification and the claims themselves where the term is used in conjunction with the conversion of data, commands, or addresses from one predefined format (such as a protocol-based channel language) to another predefined language (such as the language used by the storage facility) or to the "generic" format claimed in the '845 patent. Ex. A, 9:12-27. Therefore, "translating" in this context is properly defined as the conversion of something from one predefined format to another, predefined format.

2) "Channel Specific Format"

Relevant computer dictionary definitions of "format" are:

Source	Definition
Computer Desktop Encyclopedia	The structure or layout of an item (Ex. M)
Microsoft Computer Dictionary	In general, the structure or appearance of a unit of data (Ex. N)

"Format" Is, therefore: "the structure or layout of an item or data." Illustrative examples of formats described in the '845 patent include FIPS 60 and SCSI. Ex. A, 14:32-41.

"Channel" must be defined with reference to the specification, and it can be corroborated with a dictionary definition. Texas Digital Systems, Inc., 308 F.3d at 1203. The specification defines channel as:

> A path or link through which information passes between two devices; a channel can be either internal or external to a computer.

Ex. A, 5:58-62.

The specification further defines a "channel," as contrasted with a bus:

> The bus provides similar functions to that of the channel, except that only one controller on a particular bus can be active at one time. In channel architecture,

> which is provided on mainframe computers, all the channels may be active simultaneously.

Ex. A, 1:39-43.

Similarly, the Computer Desktop Encyclopedia defines "channel" as:

> 1. A high speed metal or optical fiber pathway between the computer and the control units of peripheral devices. Channels are used in mainframes and high-end machines. Each channel is an independent unit that can transfer data concurrently with other channels as well as the CPU. For example, in a 10-channel computer, 10 streams of data are being transmitted to and from the CPU at the same time. In contrast, the bus in a personal computer serves as a common, shared channel between all devices. Each device must wait for its turn on the bus.

(Ex. O).

The above definitions lead to a definition of the entire phrase, "channel specific format," as the structure used by a channel interface, as opposed to a bus interface, to pass information, (*i.e.* data, addresses, and commands). This definition of the whole phrase is consistent with its usage in the '845 patent. Outside of the claims, the phrase is used only twice, both times in this section describing embodiments of the claimed invention:

> Host system 12 physically attaches to a storage controller through channel 13 which uses a unique addressing scheme. The programmable storage controller 14 may communicate with multiple channel interface cards of the channels, thus allowing multiple host channels to communicate with a single storage subsystem (i.e., the combined system of programmable storage controller 14 and storage facility 16). This allows multiple assembly host architectures (e.g., the IBM 370 Class, and Unisys 2200 Class) to communicate with a single parameter storage controller. The adapter card which is in the channel of the host system receives and passes on channel program commands and requests which are configured in a channel specific format. . . . The controller emulator 214 translates the channel program commands/requests from a channel-specific format to a "generic" format including generic address information and generic requests.

Ex. A, 9:36-58. (emphasis added). Therefore, consistent with the definitions above, the '845 patent uses the phrase "channel specific format" to mean the command and addressing scheme used by the host computer to communicate over a channel (as opposed to a bus) with the programmable storage controller.

SBC attempts to construe "channel" only in connection with the preamble's use of "storage controller channels." SBC's Brief at 11-12. The cited IEEE Dictionary presents a complementary, but inconclusive, definition of "storage channel" as the [undefined] channel "that can be used to access a storage device." Id. The only help SBC's "construction" provides is to confirm that a channel is a path of access by which communication can occur.

3) Generic Format

The term "generic" appears to have no apparent special meaning to people in the computer and storage area networking arts as is indicated by the lack of a definition in the computer dictionaries and the lack of an entry in indexes of various storage area networking treatises. General dictionaries such as the New Merriam-Webster Dictionary (1989) define generic simply as "not specific, general." (Ex. P).

Absent an apparent ordinary meaning for the term "generic" as used in the context of the claims, the specification must be consulted for construction of the term. Interactive Gift Express, Inc., 256 F.3d at 1331. The claims themselves show the "generic format" is different from the "channel specific format" from which it is translated. Moreover, the specification states the storage control manager of the programmable storage controller receives the generic format and translates the generic format into a third format of addresses and requests understood by the storage facility. Ex. A, 9:58-62, item 216 in FIG 2 of the '845 patent. As it is different from the "channel specific format," the generic format must also be different from the storage facility instruction format understood by the storage facility. The claims also clearly state the generic format includes two required elements: 1) a generic address portion; and 2) a generic request or command portion. See Ex. A, 15:56-58. Therefore, there are three forms of the address (channel specific/generic/storage facility specific) and three forms of the request/command (channel specific/generic/storage facility specific) identified and employed in the claimed programmable storage controller. This is consistent with the specification. Ex. A, 9:55-65.

The claims require the generic format to be "of said programmable storage controller," requiring the generic format to be specific to the storage controller. See Ex. G at I001370 ("Applicants note that the prior art . . . fails to disclose . . . emulators that comprise means for translating . . . to a generic format of the programmable storage controller, as defined in the amended claims."). Therefore, the "generic format" is one used internally by the claimed storage controller and includes both a generic request component and a generic addressing component, both of which differ from the addressing and command formats used by the host's channel command and the targeted storage facility.[16]

C. "Means for Translating"

Inrange agrees with SBC that the following passage from claim 1 of the '845 patent is a means-plus-function element that must be construed under 35 U.S.C. §112, ¶6:

> . . . said controller emulators comprising means for translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller that includes generic address and request information.

Ex. A, claim 1, 15:52-57. Inrange also agrees that the claimed function is that which is cited directly in the claim, namely:

[16] It is possible SBC does not disagree with Inrange's construction of these elements; however, given the circular nature of the discussion in SBC's Brief, it is impossible to tell.

> translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller that includes generic address and request information.

Id. (emphasis added). Inrange disagrees, however, with SBC's cursory analysis of the structure associated with the claimed function. While SBC is correct in stating that part of the corresponding structure is software, it is incorrect in suggesting it is the only structure necessary. Moreover, it is wrong for SBC to point to broad, generalized specification language, stating that any of the elements of the invention could just as easily be implemented in firmware or hardware (Ex. A, 7:49-63), as carte blanche authority to suggest that any structure could be substituted for the disclosed, corresponding structure. By suggesting the claimed function could simply be implemented with software, hardware, or firmware, SBC has effectively covered the entire universe of possible structures, without limitations. Such an end run around the patent statutes is not allowed. In return for allowing patent applicants the flexibility to claim in means-plus-function language rather than precise structural language, Congress expressly requires the corresponding structure to be described in the specification. See 35 U.S.C. §112, ¶6. Case law interpreting that statute requires a specific, clearly linked connection between the recited structure and the claimed function. B. Braun Med., Inc., 124 F.3d at 1424. Here, the applicants provide in the specification a detailed description of the structure that corresponds to the "means for translating."

Inrange agrees with SBC that the structure at 9:55-62, discussing the general step of translation into a "generic" format, is part of the corresponding structure for the "means for translating," but it is not the only structure. The entirety of the structure corresponding to the emulator that comprises the "means for translating" is as follows[17]:

[17] In contrast to simply saying the emulator is vaguely "software" as is SBC's tactic, Inrange has identified the actual structure the software must have.

1) "A controller emulator is defined by the set of channel commands and channel programs which it interprets." In other words, the structure corresponding to the emulator depends upon the definition of each channel command and channel program. Ex. A, 8:64-67.

2) "The controller emulator must interpret each channel command and produce results consistent with the controller protocol it is emulating." As stated in the specification, the emulator structure must ultimately be "transparent" to the end user, the operator, and to the operating system hardware. Ex. A, 9:6-11.

3) "A unique and separate controller emulator is provided for each unique and separate target unit "type" specified by the channel." Ex. A, 9:12-15.

4) The emulator structure includes "a plurality of unit threads assigned to a single controller emulator." Ex. A, 9:15-18.;10:37-57 and the '845 patent FIG. 3c. The specification further describes structure for various specific embodiments, including if the host is an IBM mainframe.[18] Ex. A, 9:18-36. The threads are provided "with an interpretor which may interpret the different channel languages and/or channel dialects of the communication interface used by one or more host computers." Ex. A, 14:24-50. "A channel language is defined to be the electrical and control signaling sequences of the host interface. A channel program dialect is defined to be the ordering and context sensitive nature of the command signals within a specific channel language." Ex. A, 14:32-37. In other words, while the language is defined as a uniform standard used by many manufacturers (such as FIPS 60 or SCSI), the dialect refers to the actual implementation of that standard by a specific manufacturer such as the FIPS 60 implementation by Unisys or the FIPS 60 implementation by IBM. Therefore, threads, and the structure of the threads defined in the cited specification above, are part of the structure corresponding to the claimed function, with the described interpreters (for particular host computers) and the described emulators (for particular target units) provided for in the unit threads. Ex. A, 14:42:49.

5) As previously discussed, the emulator structure includes translating "the channel program commands/requests from a channel-specific format to a 'generic' format including generic address information and generic responses." Ex. A, 9:55-62. The disclosed emulators translate both address and request information into their corresponding generic formats.

Together, these five general structures (which have additional specific structure as described in the specification depending upon their implementation) form the structure corresponding to a "means for translating."

[18] To the extent more specific embodiments become relative to this case, the specification's recited structure corresponding to these more specific embodiments becomes the structure for performing the claimed function.

D. A "First Interface" or "Channel Programs Transmitted from the Channels of the Host"

Claim 1 of the '845 patent claims a storage control subsystem comprising a:

> first interface for interfacing a plurality of channel adapters which carry a plurality of channel programs transmitted from the channels of the host system to said programmable storage controller, each channel program having means for carrying data, status information, and commands.

Claim 34 claims a storage controlling method comprising:

> receiving a plurality of channel programs transmitted from the channels of the host system, to said programmable storage controller, each channel program carrying at least one of data, status information and commands.

In claim 1, the "first interface" receives channel commands from an existing channel already coming from the host and provides these channel commands to the programmable storage controller. Ex. A, 15:42-47. Similarly, in claim 34, the channel programs are received from the preexisting channels of the host system. Ex. A, 18:5-9. Once again, SBC presents a circular and curt construction of this element stating the claim means what it says (whatever that may be). Moreover, SBC's definition of the first interface is unclear as to whether (i) host bus adapters (HBAs) are a first interface and bus communications are channels (both incorrect statements), or (ii) HBAs and buses are distinct elements from the first interface and the host channels.

As previously noted, a channel is not a bus. Ex. A, 1:39-43. Allen Eng'g Corp., 299 F.3d at 1344. The HBA provides an adapter that receives bus communications and converts them to channel programs that can be transmitted over the channel. Without the HBA, there would be no channel communications from the host computer. Since the first interface must receive these channel programs from the channel, it would be impossible for the HBA residing on a host computer's system bus to form part of the first interface. By its very nature, the HBA does not receive a channel command from the host, it resides on the system bus of the host computer and receives something different – a bus communication. A first interface of the "storage control subsystem" is therefore distinct from a HBA because the first interface is instead configured to interface the programmable storage controller with a plurality of channel adapters. Similarly, receiving a plurality of channel programs transmitted from the channels of the host cannot occur at the host bus adapter of the host computer, but must occur after the host bus adapter creates those channel programs and transmits them, over the channel, to the first interface. Instead, the claimed "first interface" is:

> An interface that connects the programmable storage controller to the plurality of channel adapters, where the first interface excludes the host bus adapter(s).

E. "Channel Program Having Means For Carrying Data, Status Information and Commands"

The element "channel program having means for carrying data, status information and commands uses the means-for format without reciting additional structure; therefore it is a means-plus-function element governed by § 112, ¶6. While "channel programs" may have the characteristics noted in the specification sections cited by SBC (SBC's Brief at 16-17), they must also have the claimed function: "carrying data, status information and commands," and they must further have the claimed function's corresponding structure or its equivalent as recited in the specification. SBC completely ignores the fact that this element is in the means-plus-function format and therefore erroneously construes the term.

The structure corresponding to the channel program's means for carrying data, status information and commands is: a unique dialect often referred to in the patent as a channel-specific format. See Ex. A, 2:1-6 (". . .the channel program command set will have a unique "dialect."); 14:32-40 ("A channel program dialect is defined to be the ordering and context sensitive nature of the command signals within a specific channel language."); 8:16-21 (". . .

based on the particular device (target unit) requested by the host system through the channel program"); 9:47-48 ("... channel program commands and requests which are configured in a channel specific format")[19]; 9:55-58 ("The controller emulator 214 translates the channel program commands/requests from a channel-specific format to a "generic" format including generic address information and generic requests.").

The construction of this claim term should therefore be limited to the corresponding structure cited above, and statutory equivalents thereof.

F. "Computer" and "Application Program"

In the asserted claims, it is clear that the required "programmable storage controller" be "implemented with an application program and a computer." Ex. A, 15:37-8. (emphasis added) By virtue of its own prosecution history disclaimer and the unchallenged Examiner's Statement of Reasons for Allowance, the "computer" required by the claims is limited to a general purpose computer:

> prior arts of record do not teach a programmable storage controller that emulates a plurality of types of target unit specific storage controllers. This programmable storage controller is implemented with application programs and a general-purpose computer that allows for the user to switch between these application programs without reloading or changing the operating system, or physically modifying the hardware configuration. This programmable storage controller, further, comprises a plurality of controller emulators for translating channel programs and commands from a channel specific format [address and requested information] to a generic format.
>
> The examiner considers the applicants' claims . . . to be allowable based on the claim interpretation and the aforesaid prior arts of record.

[19] "For specific examples of Direct and Sequential storage device channel command and program definitions which the controller emulators may be configured to interpret, please refer to IBM 3990 Storage Control Reference (GA32-0099-3) and IBM 3480 Magnetic Tape Subsystem Reference: Channel Commands, Status and Sense Bytes, and Error Recovery Procedures (GA32-0042-6) . . ." 9:28-35.

Ex., I001354-55 (emphasis added, bracketed text in original); Southwall Tech., Inc., 54 F.3d at 1576. The patent itself defines a "general purpose computer" as "a computer which is provided with enough facilities to allow it to implement a wide range of different unrelated operating systems and/or applications." Ex. A, 5:42-7. Therefore, the "computer" required in each of the asserted claims must be a "computer which is provided with enough facilities to allow it to implement a wide range of different unrelated operating systems and/or applications."

SBC tries to evade the prosecution history with a conclusory assertion that while the claims of the parent application did require a general purpose computer, such a requirement was dropped in the claims of the File Wrapper Continuation Application. SBC's Brief at 14.[20] SBC's assertion is contrary to the law. A patentee that disclaims a claim's scope during the prosecution of the patent, even where the disclaimer is in a parent application, cannot recapture that scope in a later child application. Wang Labs.,197 F.3d at 1384. As previously noted, although deemed an ineffective argument by the examiner, no less than three times the applicants sought allowance of proposed claims by arguing its patent was distinguished from the prior art because the applicants' were claiming a general purpose computer. Ex. G, I001430; I001432-46; I001403-19; Bell Atlantic Network Servs., Inc., 262 F.3d at 1268-1273.

Finally, the Examiner's Statement of Reasons for Allowance (which the patentee did not dispute) easily overcomes any claim differentiation argument SBC is expected to assert by virtue of the fact that the term "general purpose computer" appears in other claims of the '845

[20] SBC's reliance on Rambus Inc. v. Infineon Technologies AG, 318 F.3d 1081, 1090 (Fed. Cir.2003) and Intervet America, Inc. v. Kee-Vet Labs., Inc., 887 F.2d 1050, 1053-54 (Fed. Cir. 1989) is misplaced. These cases involved passing remarks in the prosecution history that the court held did not add elements to the claims, not a clear and repeated disavowal of claim scope stated explicitly in attempts to overcome the prior art and ultimately relied upon by the examiner as a reason for granting the patent. In fact, Intervet America, Inc., explicitly distinguished its holding from a situation where "the attorney had represented to the examiner [the disclaimed subject matter] was *not* intended to be included in order to get the claim allowed." 887 F.2d at 1054 (In such situations, "the patentee may be estopped to contend otherwise.") (emphasis in original).

patent. It is well settled that the doctrine of claim differentiation creates only a presumption that each claim has a different scope; it is not a 'hard and fast' rule of construction. See Bristol-Myers Squibb Co. v. Ben Venue Labs. Inc., 246 F.3d 1368, 1376 (Fed. Cir. 2001); Multiform Desiccants, Inc. v. Medzam, Ltd., 133 F.3d 1473, 1480 (Fed. Cir. 1998) (affirming the district court's construction of a claim although it rendered a dependent claim redundant in light of the specification and file history). Therefore, the Examiner's decision to allow the patent on the explicit condition that all of the claims be construed to require a "general purpose computer" clearly and unambiguously controls the meaning of the term "computer" in claims 1 and 34. Elkay Mfg. Co., 192 F.3d at 980. Ex. G EI001354 ("This programmable storage controller is implemented with application programs and a general-purpose computer...") (emphasis added).

In fact, for the claims that issued to be valid, the examiner gave weight to his specific understanding of the claims by stating: "The examiner considers the applicants' claims . . . allowable based on the claim interpretation and the aforesaid prior arts of record." Ex. G, I001354 (emphasis added). When the applicant had an opportunity to correct, modify, or even comment upon the examiner's Statement of Allowance, the applicant affirmed, rather than disputed, the examiner's interpretation.

> Applicants wish to clarify the record with respect to the basis for patentability of claims in the present application . . . Applicants do not disagree with the Examiner's indication that certain features are not disclosed or taught by the prior art, as indicated by the Examiner in the State of Reasons for Allowance . . . .

Ex. G, I001336-7. The applicant therefore clearly disclaimed using anything but a general purpose computer, they carried that requirement through to the issued claims, and they received the '845 patent with the examiner clearly stating the computer was limited to a general purpose computer. SBC cannot now hope to regain the claim scope previously abandoned on the public record.

SBC has only summarily construed "application program" as: "prescribed rules for operating a computer." SBC's Brief at 15. Here, SBC's effort is not inconsistent, it is just incomplete. First, the applicants performed no "lexicography" on the term "application program," so SBC is wrong to equate the '845 patent's definition of "computer" with a special definition of "application program." See Ex. A, 5:26-6:15.[21] Second, SBC cites to no technical subject matter dictionaries to support its broad "rules for operating a computer" definition of this term.[22] Consequently, in the context of the '845 patent and file history, the concept of an "application program" has the particular meaning:

> software code that is compatible with the software operating system configured to run a general purpose computer and that is capable of performing particular applications on the computer without changing the operating system or modifying the hardware configuration.

Inrange's construction is consistent with the applicants' statements during prosecution history description. See Ex. G, I 001378, 1382-1385 (describing the "general purpose computer" and its attributes). This construction is also consistent with the concept of a "programmable" device as understood by the examiner as the basis for allowance. See cited Notice of Allowance,

[21] Indeed, in the definition of "general purpose computer" given in the '845 patent, the applicants acknowledged that it implements a range of different unrelated operating systems and/or applications." Ex. A, 5:42-46.

[22] As explained in the Computer Desktop Encyclopedia, an application program is:

> Any data entry, update, query or report program that processes data for the user. It includes the generic productivity software (spreadsheets, word processors, database programs, etc.), as well as custom and packaged programs for payroll, billing, inventory and other accounting purposes. ...Contrast with system program.

(Ex. Q).
The same source defines "system software" as:

> Programs used to control the computer and develop and run application programs. It includes operating systems, TP monitors, network operating systems and database managers. Contrast with *application program*.

(emphasis in original) (Ex. R).

above. This construction is also consistent with and supported by the '845 patent specification. See Ex. A, 7:50-53 ("programmable storage controller (14) is implemented by means of software code used to configured and run computer (15)."); compare 13:43-52 (describing an alternative embodiment that has a bare essentials "specialized" operating system on a mini-processor, so that only an application program and essential operating system services are running).

The "application program" is associated with and defined by its role in a "general purpose computer" and it should not be broadened to encompass any or all "rules for operating a computer."

G. Target Unit

SBC and Inrange agree on this claim term's construction. "target unit" or "target units" means and refers to the storage devices that together make up the storage facilities. See SBC's Brief at 20. The specification makes clear that target units has this meaning. See Ex. A, Abstract: the second interface "allows input and output to storage facilities which comprise one or more target units"; 4:11-12 (different kinds of storage devices comprise the storage facilities); 5:6-9 ("The target units may comprise random access and sequential storage devices . . . .").

Nothing the record indicates a different construction for the term "target unit."

IV. CONCLUSION

SBC has offered nothing more than superficial, and often circular, claim constructions for elements in need of construction in this case. It is not enough to wait and tell a lay jury only that the ordinary meaning (to one skilled in the art) should apply to technical claim terms. The entire purpose of rendering claim construction as a matter of law is eviscerated when the meanings of disputed claims, even those that require only their ordinary meaning, are not decided in advance of trial by the court. Inrange has presented this Court with a complete and accurate construction of elements of asserted claims in dispute.

To further the process and respond to any issues relating to claim construction, Inrange respectfully requests the Court provide an opportunity for an oral hearing on the issues presented in the parties' respective briefs.

Respectfully submitted,

INRANGE TECHNOLOGIES CORP.

Dated: October 26, 2004.

By: Kevin D. Conneely
Kevin D. Conneely (MN Bar No. 192703)
Admitted Pro Hac Vice

LEONARD, STREET & DEINARD
Professional Association
150 South Fifth Street
Suite 2300
Minneapolis, MN 55402
Tel: (612) 335-1500
Fax: (612) 335-1657

AND

William T. Reid, IV
DIAMOND, MCCARTHY, TAYLOR, FINLEY,
BRYANT & LEE, LLP
6504 Bridge Point Parkway
Suite 400
Austin, TX 78730
Tel: (512) 617-5205
Fax: (512) 617-5299

AND

Benjamin King
DIAMOND MCCARTHY, TAYLOR, FINLEY,
BRYANT & LEE, LLP
Renaissance Tower
1201 Elm Street, 34th Floor
Dallas, TX 75270
Tel: (214) 389-5300
Fax: (214) 389-5399

ATTORNEYS FOR INRANGE TECHNOLOGIES CORPORATION

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF TEXAS
DALLAS DIVISION

SBC TECHNOLOGY RESOURCES, INC.,

Plaintiff,

vs.

INRANGE TECHNOLOGIES CORP.,
ECLIPSYS CORP.; and
RESOURCE BANCSHARES
MORTGAGE GROUP, INC.,

Defendants.

HONORABLE David C. Godbey

CIVIL ACTION NO. 303-CV-418-N

DEFENDANT INRANGE TECHNOLOGIES CORPORATION'S CLAIM CONSTRUCTION APPENDIX

TAB	DESCRIPTION
A	United States Patent No. 5,530,845
B	Excerpt from Distributed Storage Networks: Architecture, Protocols and Management by Thomas C. Jepsen
C	SNIA definition of "storage area network"
D	Excerpt from Storage Networks: The Complete Reference by Robert Spalding
E	SNIA definition of "RAID"
F	Excerpt from Designing Storage Area Networks, 2nd ed by Tom Clark
G	Patent prosecution binder
H	United States Patent No. 4,803,623 (Klashka)
I	The New Merriam-Webster Dictionary definition of "patent"
J	SNIA definition of "switch"
K	The Computer Desktop Encyclopedia (Freedman) definition of "translate"
L	Microsoft Computer Dictionary, 4th ed. definition of "translate"
M	The Computer Desktop Encyclopedia (Freedman) definition of "format"
N	Microsoft Computer Dictionary, 4th ed. definition of "format"
O	The Computer Desktop Encyclopedia (Freedman) definition of "channel"
Q	The New Merriam-Webster Dictionary definition of "generic"
R	The Computer Desktop Encyclopedia (Freedman) definition of "application program"
S	Phillips v. AWH Corp., 376 F.3d 1382

Respectfully submitted,

INRANGE TECHNOLOGIES CORP.

Dated: 10/26/04.

By: Kevin D. Conneely
Kevin D. Conneely (MN Bar No. 192703)
Admitted Pro Hac Vice

LEONARD, STREET & DEINARD
Professional Association
150 South Fifth Street
Suite 2300
Minneapolis, MN 55402
Tel: (612) 335-1500
Fax: (612) 335-1657

AND

William T. Reid, IV
DIAMOND, MCCARTHY, TAYLOR, FINLEY,
BRYANT & LEE, LLP
6504 Bridge Point Parkway
Suite 400
Austin, TX 78730
Tel: (512) 617-5205
Fax: (512) 617-5299

AND

Benjamin King
DIAMOND MCCARTHY, TAYLOR, FINLEY,
BRYANT & LEE, LLP
Renaissance Tower
1201 Elm Street, 34th Floor
Dallas, TX 75270
Tel: (214) 389-5300
Fax: (214) 389-5399

Attorneys for InRange Technologies Corporation

A

US005530845A

United States Patent [19]

Hiatt et al.

[11] Patent Number: 5,530,845

[45] Date of Patent: Jun. 25, 1996

[54] STORAGE CONTROL SUBSYSTEM IMPLEMENTED WITH AN APPLICATION PROGRAM ON A COMPUTER

[75] Inventors: David M. Hiatt; Timothy R. Klos, both of St. Louis, Mo.

[73] Assignee: Southwestern Bell Technology Resources, Inc., St. Louis, Mo.

[21] Appl. No.: 373,896

[22] Filed: Jan. 17, 1995

Related U.S. Application Data

[63] Continuation of Ser. No. 882,010, May 13, 1992, abandoned.

[51] Int. Cl.^{6} G06F 13/10

[52] U.S. Cl. 395/500; 395/882; 395/883

[58] Field of Search 395/500, 882, 395/883

[56] References Cited

U.S. PATENT DOCUMENTS

3,544,969	12/1970	Rakoczi et al. .	
4,084,235	4/1978	Hirtle et al. .	
4,277,827	7/1981	Carlson et al. .	
4,394,732	7/1983	Swenson .	
4,394,733	7/1983	Swenson .	
4,399,503	8/1983	Hawley .	
4,415,969	11/1983	Bayliss et al. .	
4,425,615	1/1984	Swenson et al. .	
4,454,595	6/1984	Cage .	
4,467,421	8/1984	White .	
4,476,526	10/1984	Dodd .	
4,500,933	2/1985	Chan .	
4,511,963	4/1985	Kantner .	
4,638,423	1/1987	Ballard .	
4,775,969	10/1988	Osterlund .	
4,792,896	12/1988	MacLean et al. .	
4,803,623	2/1989	Klashka et al.	395/275
4,812,975	3/1989	Adachi et al. .	
4,855,905	8/1989	Estrada et al. .	
4,864,291	9/1989	Korpi .	
4,864,532	9/1989	Reeve et al. .	
4,868,734	9/1989	Idleman et al. .	
4,888,691	12/1989	George et al. .	
4,888,727	12/1989	Getson, Jr. et al. .	
4,965,801	10/1990	Dulac .	
5,088,033	2/1992	Binkley et al. .	
5,131,082	7/1992	Bonevento et al. .	
5,151,985	9/1992	Sander et al. .	
5,247,638	9/1993	O'Brien et al.	395/425

FOREIGN PATENT DOCUMENTS

246125	3/1994	Argentina .

OTHER PUBLICATIONS

Copy of a diagram, for an adaptor to link an ECKD/byte multiplexer channel to Micro Channel.

Primary Examiner—Krisna Lim
Attorney, Agent, or Firm—Greenblum & Bernstein

[57] ABSTRACT

A storage controller is disclosed which may emulate several types of specialized host specific and/or storage device specific storage controllers. The storage controlling system can transfer information between one or more different types of target units and one or more channels of at least one host. The system is provided with a computer, which includes a first interface, a second interface, and a programmable storage controller. The first interface is configured to receive one or more channel adapters which carry one or more channel programs transmitted from the channels of the host. The channel programs may carry data, status information, and commands. The second interface allows input and output to storage facilities which comprise one or more target units. The programmable storage controller may be provided with a device coupled to the channel adapters for translating channel program commands, and determining, from the channel program, a target unit for which at least one channel program is transmitted. A set of equipment controllers is provided which interpret channel program commands and status information, and which further control data transfers to and from the storage facilities in accordance with the channel program command. A device is also provided for establishing a unit thread by choosing an equipment controller from the set of equipment controllers as a function of the type of equipment that the channel requests as a target.

48 Claims, 8 Drawing Sheets

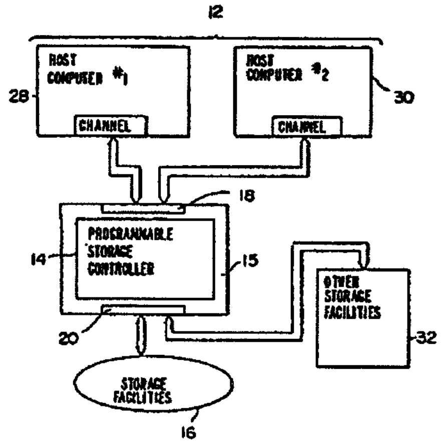


Fig- 1

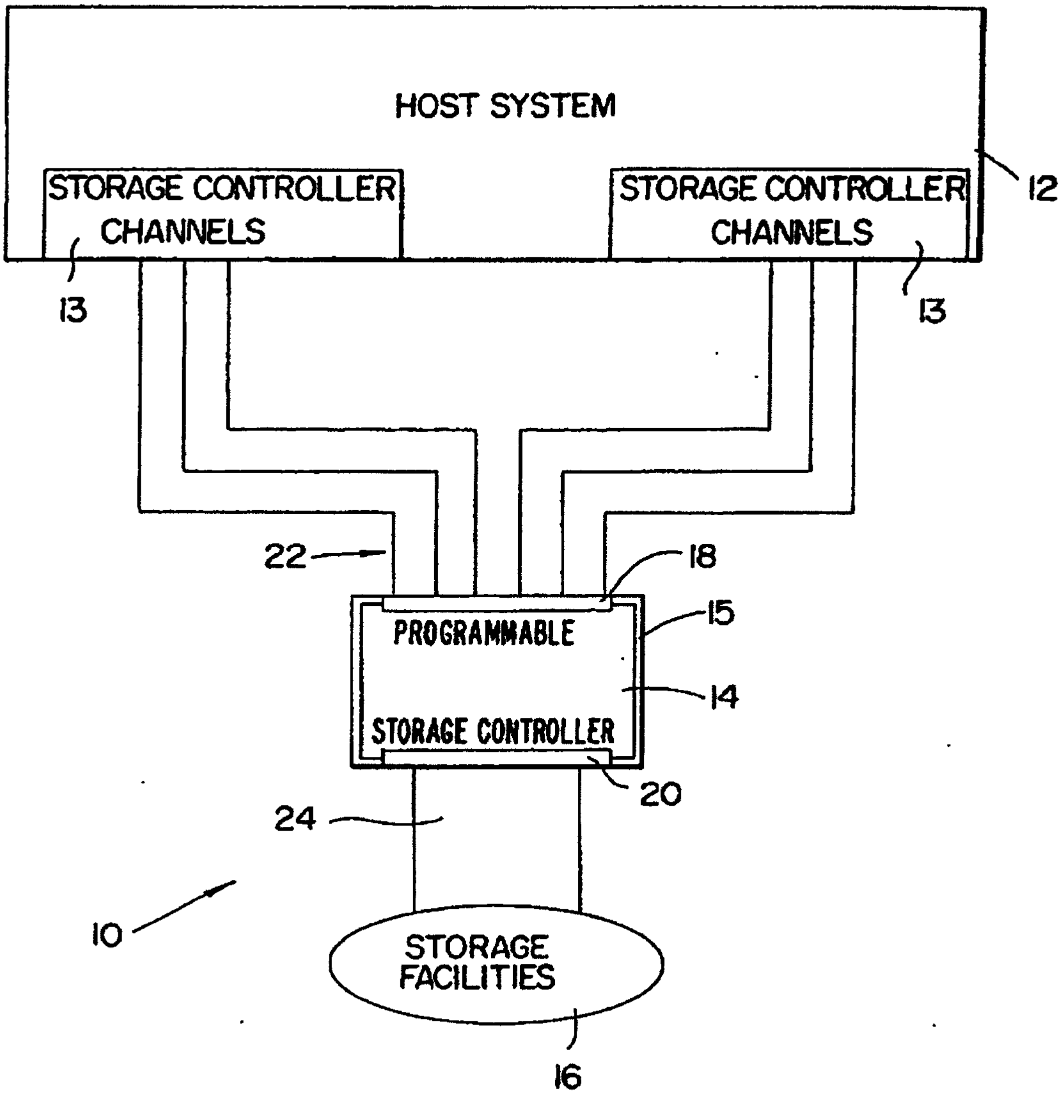

HOST SYSTEM
STORAGE CONTROLLER CHANNELS
STORAGE CONTROLLER CHANNELS
12
13
13
22
18
15
PROGRAMMABLE
14
STORAGE CONTROLLER
20
24
10
STORAGE FACILITIES
16

Fig- 2

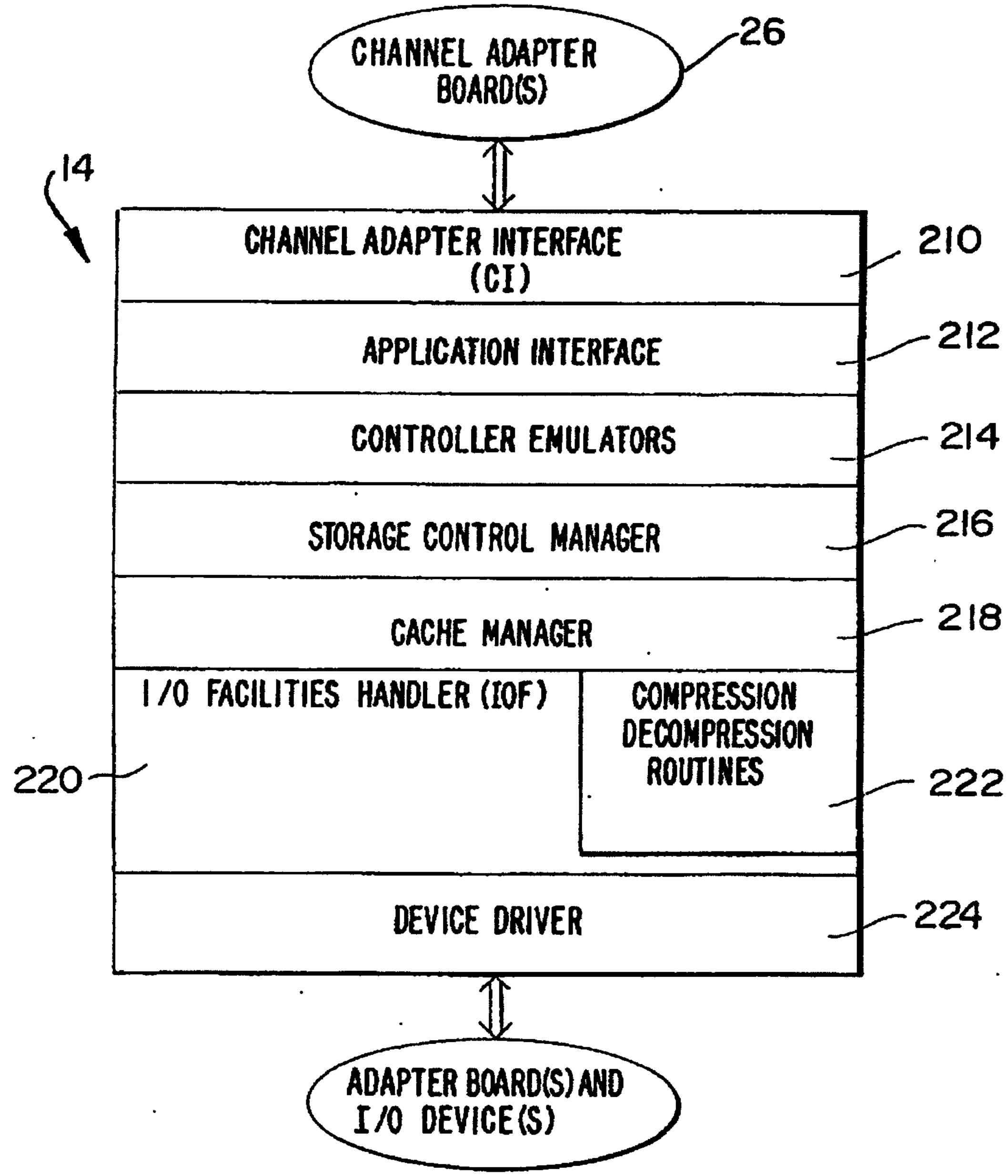
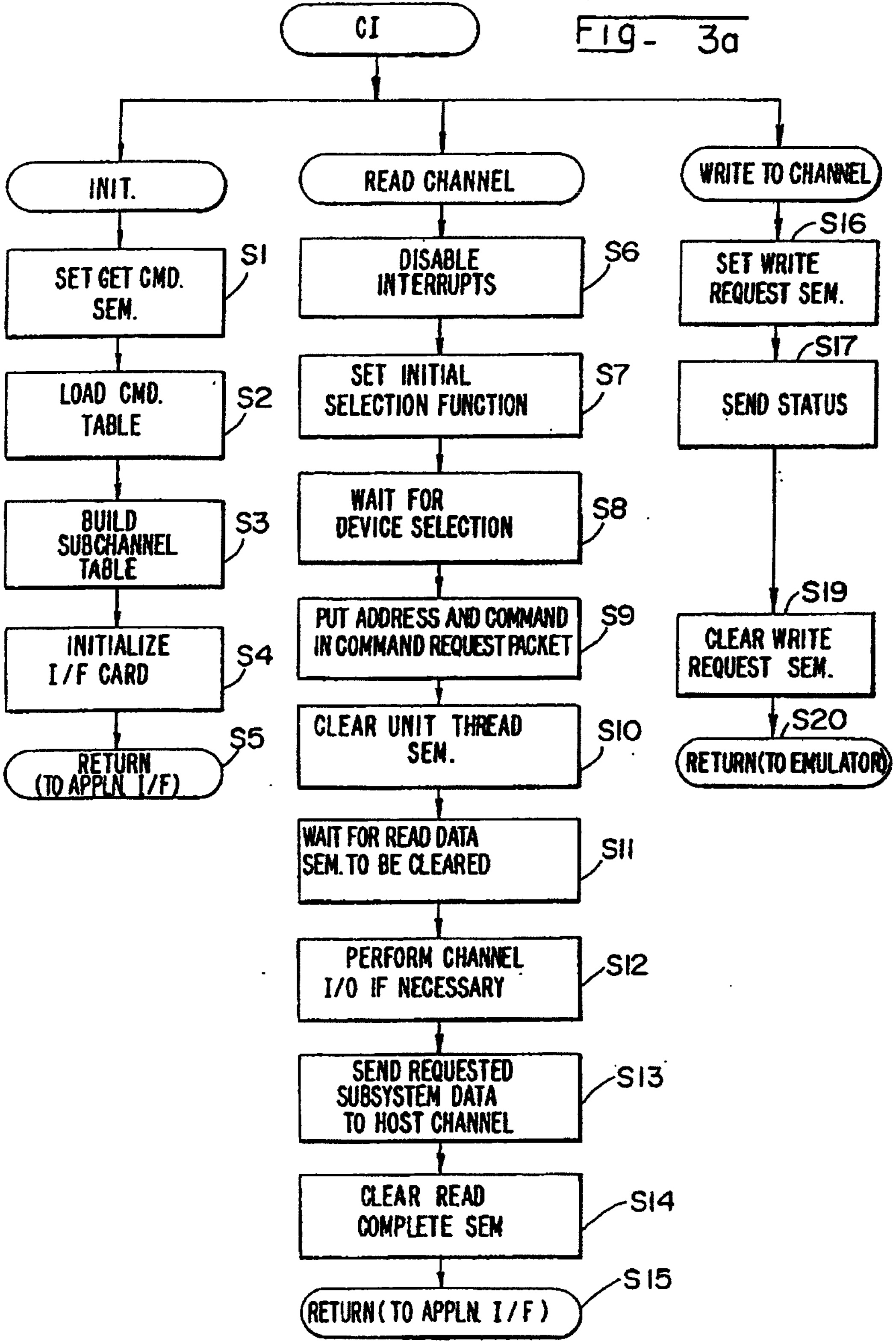

CHANNEL ADAPTER BOARD(S)
26
14
CHANNEL ADAPTER INTERFACE (CI)
210
APPLICATION INTERFACE
212
CONTROLLER EMULATORS
214
STORAGE CONTROL MANAGER
216
CACHE MANAGER
218
I/O FACILITIES HANDLER (IOF)
220
COMPRESSION DECOMPRESSION ROUTINES
222
DEVICE DRIVER
224
ADAPTER BOARD(S) AND I/O DEVICE(S)

Fig- 3a


CI
INIT.
SET GET CMD. SEM.
S1
LOAD CMD. TABLE
S2
BUILD SUBCHANNEL TABLE
S3
INITIALIZE I/F CARD
S4
RETURN (TO APPLN I/F)
S5
READ CHANNEL
DISABLE INTERRUPTS
S6
SET INITIAL SELECTION FUNCTION
S7
WAIT FOR DEVICE SELECTION
S8
PUT ADDRESS AND COMMAND IN COMMAND REQUEST PACKET
S9
CLEAR UNIT THREAD SEM.
S10
WAIT FOR READ DATA SEM. TO BE CLEARED
S11
PERFORM CHANNEL I/O IF NECESSARY
S12
SEND REQUESTED SUBSYSTEM DATA TO HOST CHANNEL
S13
CLEAR READ COMPLETE SEM
S14
RETURN (TO APPLN. I/F)
S15
WRITE TO CHANNEL
S16
SET WRITE REQUEST SEM.
S17
SEND STATUS
S19
CLEAR WRITE REQUEST SEM.
S20
RETURN (TO EMULATOR)

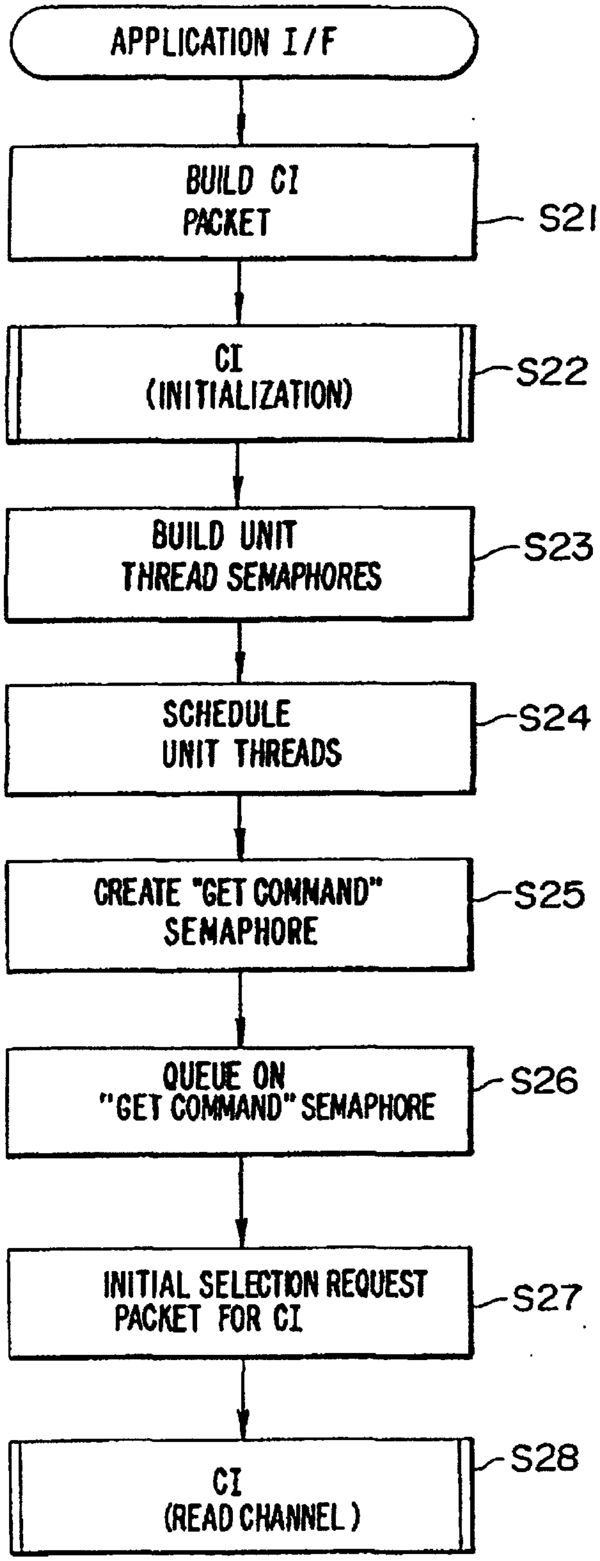

APPLICATION I/F
BUILD CI PACKET
S21
CI (INITIALIZATION)
S22
BUILD UNIT THREAD SEMAPHORES
S23
SCHEDULE UNIT THREADS
S24
CREATE "GET COMNAND" SEMAPHORE
S25
QUEUE ON "GET COMMAND" SEMAPHORE
S26
INITIAL SELECTION REQUEST PACKET FOR CI
S27
CI (READ CHANNEL)
S28

Fig - 3b

Fig- 3c

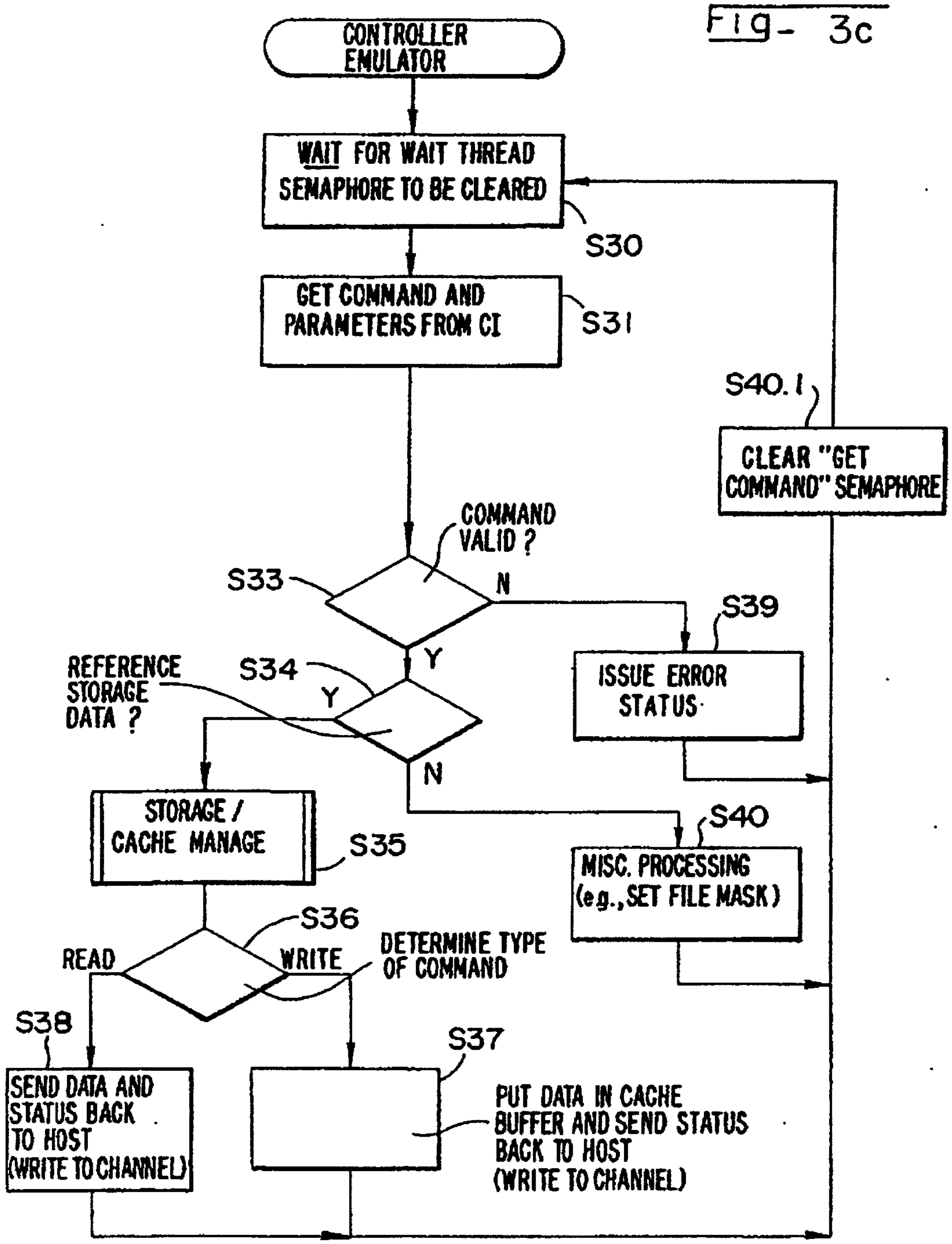

CONTROLLER EMULATOR
WAIT FOR WAIT THREAD SEMAPHORE TO BE CLEARED
S30
GET COMMAND AND PARAMETERS FROM CI
S31
S40.1
CLEAR "GET COMMAND" SEMAPHORE
COMMAND VALID ?
S33
N
Y
S39
ISSUE ERROR STATUS
REFERENCE STORAGE DATA ?
S34
Y
N
S40
MISC. PROCESSING (e.g., SET FILE MASK)
STORAGE / CACHE MANAGE
S35
S36
READ
WRITE
DETERMINE TYPE OF COMMAND
S38
SEND DATA AND STATUS BACK TO HOST (WRITE TO CHANNEL)
S37
PUT DATA IN CACHE BUFFER AND SEND STATUS BACK TO HOST (WRITE TO CHANNEL)

Fig - 3d

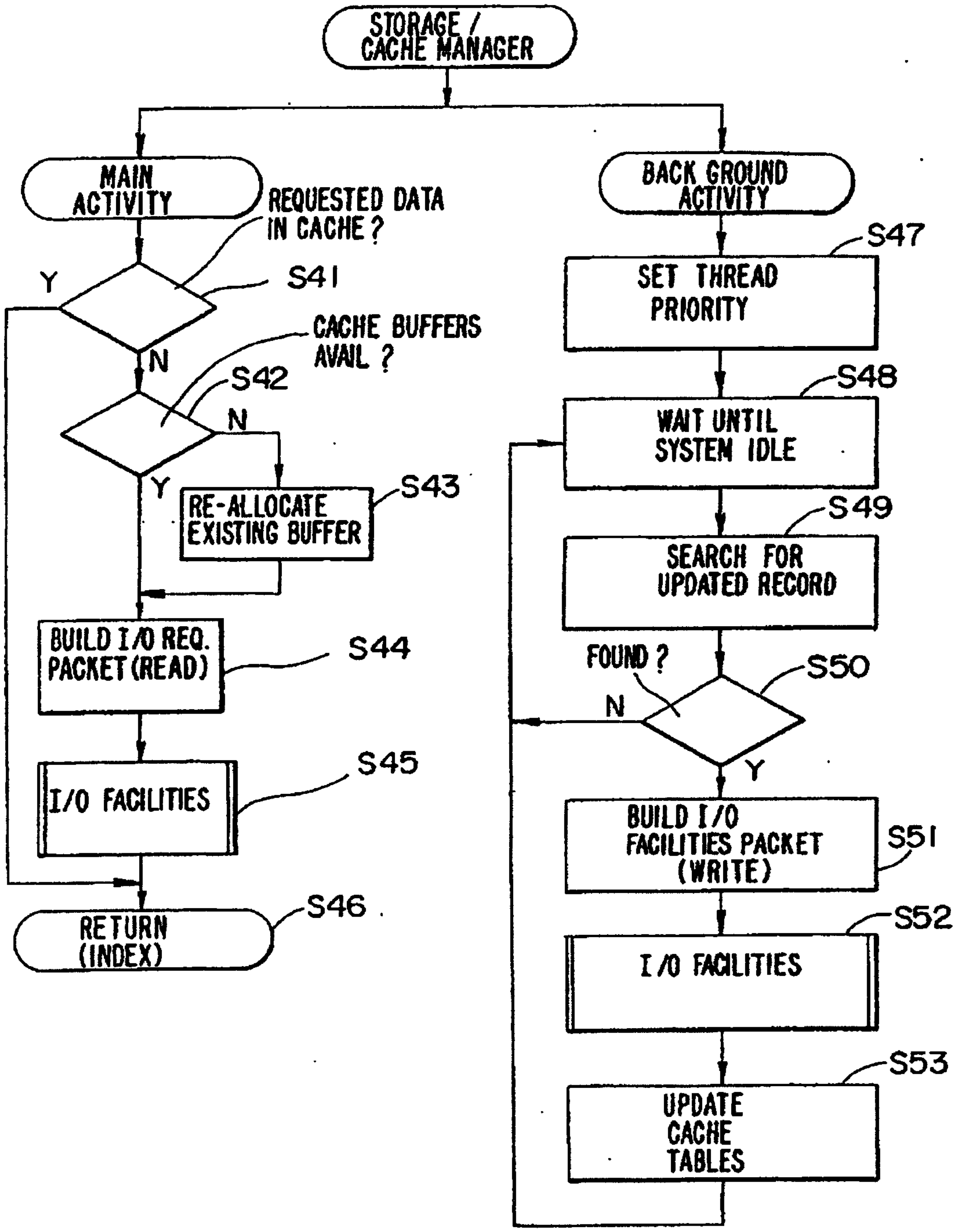

STORAGE / CACHE MANAGER
MAIN ACTIVITY
REQUESTED DATA IN CACHE ?
S41
Y
N
CACHE BUFFERS AVAIL ?
S42
N
Y
RE-ALLOCATE EXISTING BUFFER
S43
BUILD I/O REQ. PACKET (READ)
S44
I/O FACILITIES
S45
RETURN (INDEX)
S46
BACK GROUND ACTIVITY
S47
SET THREAD PRIORITY
S48
WAIT UNTIL SYSTEM IDLE
S49
SEARCH FOR UPDATED RECORD
FOUND ?
S50
N
Y
BUILD I/O FACILITIES PACKET (WRITE)
S51
S52
I/O FACILITIES
S53
UPDATE CACHE TABLES

Fig- 4

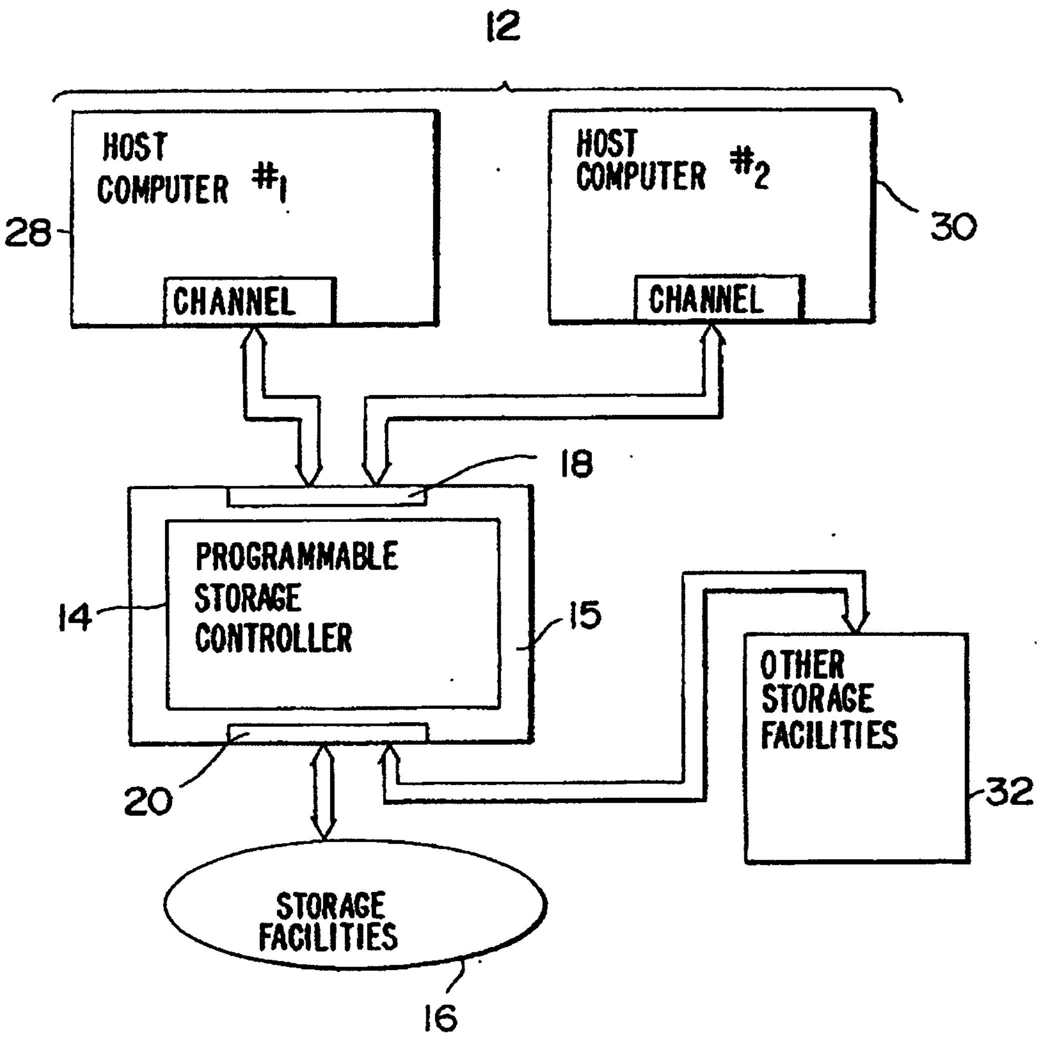
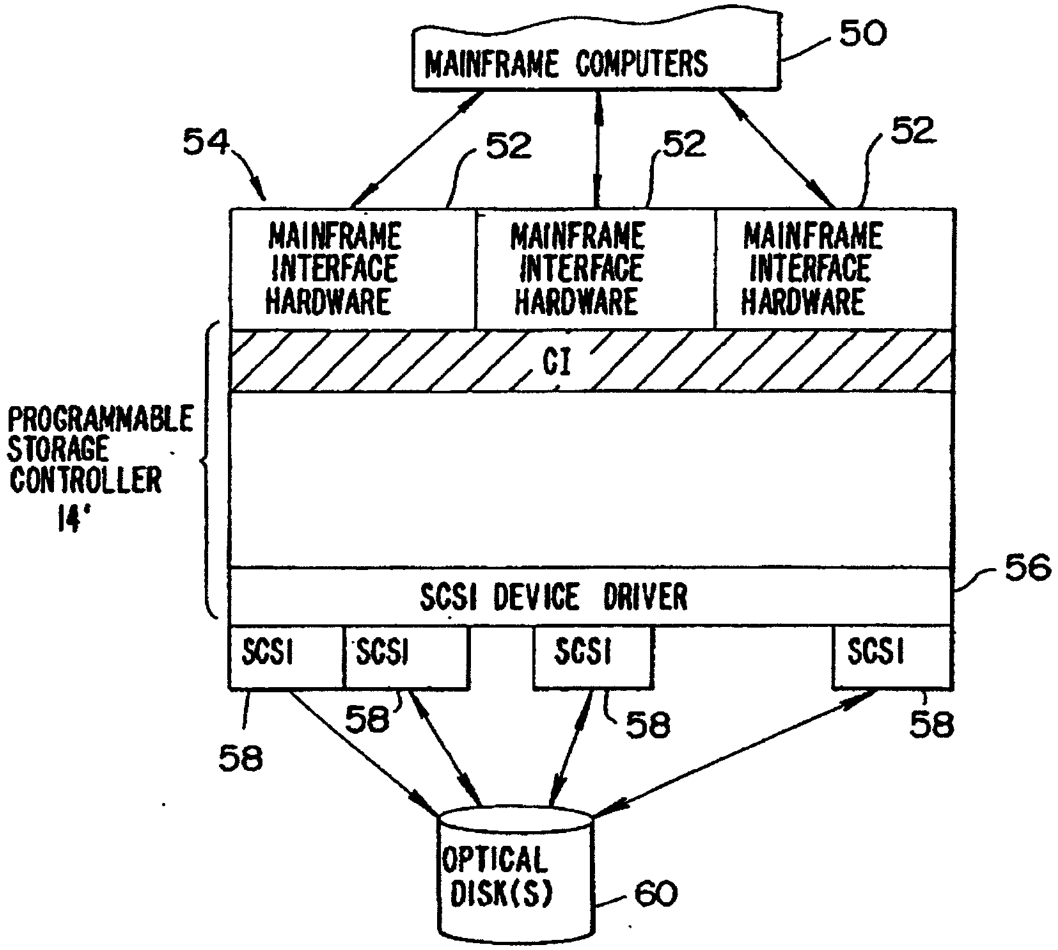

12
HOST COMPUTER #1
CHANNEL
28
HOST COMPUTER #2
CHANNEL
30
18
PROGRAMMABLE STORAGE CONTROLLER
14
15
OTHER STORAGE FACILITIES
32
20
STORAGE FACILITIES
16

Fig- 5


MAINFRAME COMPUTERS
50
54
52
52
52
MAINFRAME INTERFACE HARDWARE
MAINFRAME INTERFACE HARDWARE
MAINFRAME INTERFACE HARDWARE
CI
PROGRAMMABLE STORAGE CONTROLLER 14'
SCSI DEVICE DRIVER
56
SCSI
SCSI
SCSI
SCSI
58
58
58
58
OPTICAL DISK(S)
60

STORAGE CONTROL SUBSYSTEM IMPLEMENTED WITH AN APPLICATION PROGRAM ON A COMPUTER

This application is a continuation of application Ser. No. 07/882,010, filed May 13, 1992, now abandoned.

FIELD OF THE INVENTION

The present invention relates to a storage controller for a host computer system. More particularly, the present invention relates to a programmable storage controller implemented on a computer which is capable of emulating one or more storage device specific and/or host specific storage controllers.

DISCUSSION OF BACKGROUND INFORMATION

Mainframe computer systems, such as the IBM 3990, the IBM 4381, and the UNISYS 2200/200, send and retrieve data to and from storage facilities via storage controllers. Such mainframe computers communicate with their peripheral storage facilities, such as disk and tape systems, via one or more "channels." These channels carry commands and data between the computer and the storage facilities. The channels extend between, or "bridge" the gap between, the mainframe computer's main processor (i.e., the CPU) and the storage controller. The storage controller then interprets the commands and manipulates the storage facilities to satisfy a request.

Mini computers also communicate with storage facilities via storage controllers. Normally, storage controllers for mini computers are far less sophisticated than those for mainframe computers; however, they provide essentially the same, albeit more limited, services or functions. For example, in mini computer systems, storage controllers are typically held on a common bus, rather than on separate channels for each string of disks or other storage devices. The bus provides similar functions to that of the channel, except that only one controller on a particular bus can be active at one time. In channel architecture, which is provided on mainframe computers, all the channels may be active simultaneously. Thus, mainframe computers have much wider maximum I/O band widths than mini computers.

At the present time, due to rapid advances in peripheral technologies, newly developed storage facilities are available which have increased capabilities in areas such as efficiency and size. For example, rewritable optical disks, optical tapes, and 4 mm digital tapes (DAT) are each known for their large storage capacities, reasonably quick access time, and low floor space and power requirements. However, in order for current mainframe and mini computers to access these improved storage facilities, specialized storage controllers must be developed (or purchased) which may handle such storage facilities. For example, in order to facilitate connection of mainframe FIPS 60 channels to a SCSI storage device interface, several specialty manufacturers provide plug-in boards which allow VME base computers, such as the SUN, to intercept FIPS 60 channel inputs. However, these products have been provided by small organizations for limited specialty applications; they are not generic storage controllers which will support all or a significant portion of newly developed storage facilities.

Another significant limitation of conventional storage controllers is that they are typically limited in their ability to communicate with only the specific operating system of one host computer system. Although many mainframe channels utilize the FIPS 60 channel communication protocol, for each unique operating system (e.g., UNIX, OS/2 DOS, AIX), the channel program command set will have a unique "dialect." Thus, "host-specific" storage controllers must be provided to support each operating system.

There is a tremendous cost associated with buying or developing specialized (host specific and/or storage device specific) storage controllers. Thus, there is a need for a generalized, versatile, programmable storage controlling system which would allow various host computer systems to utilize, and thus benefit from, the increased advantages of new peripheral storage facilities that are now available, or which will be available in the near future.

SUMMARY OF THE INVENTION

In view of the above, the present invention, through one or more of its various aspects and embodiments, is thus intended to bring about one or more of the following objects and advantages.

One object of the present invention is to provide a single storage controller which may emulate several types of specialized host specific and/or storage device specific storage controllers. A further object of the present invention is to allow free substitution of storage devices, and thus provide a storage controller which will control data transfer to and from different types of both sequential and random access storage devices.

It is yet a further object of the present invention is to provide a storage controlling system which has an intercontroller communication bridge, which allows controller facilities to be shared.

Still a further object of the present invention is to provide a storage controlling system which will be capable of archiving data directly without intervention or use of the host computer system.

It is yet a further object of the present invention to integrate new peripheral storage facility technology, independently of the requirements and/or limitations of the host computer system.

It is yet a further object of the present invention to provide a storage controlling system which has customized controller services such as data compression and caching algorithms.

It is yet a further object of the present invention to provide a storage controlling system which may be utilized simultaneously by a number of host computers, wherein the host computers may have different operating systems and/or different channel communications protocols.

It is yet a further object of the present invention to substantially reduce the number of controllers required in channel architecture computers, by providing a programmable general purpose storage controlling system.

It is yet a further object of the present invention to provide a storage controlling system which has enhanced efficiency characteristics, such as dynamic data compression.

It is a further object of the present invention to provide a storage controlling system which can achieve greater storage capacities, lower costs, lower power consumption, and less use of floor space, by allowing a host computer system to be connected to a variety of newly available storage facility technologies.

The present invention, therefore, is directed to a storage controlling system for transferring information between one or more target units and one or more channels of at least one host. The host is configured for one or more types of equipment corresponding to the one or more target units. The system is provided with a computer, which includes a first interface, a second interface, and a programmable storage controller. The first interface is configured to receive one or more channel adapters which carry one or more channel programs transmitted from the one or more channels of the host. Each channel program typically carries data, status information and commands. The second interface interfaces, and thus allows input and output, to storage facilities which comprise the one or more target units. In a particular aspect, the type of equipment for which the host is configured is different than the one or more target units.

In accordance with a particular aspect of the present invention, the programmable storage controller comprises a device coupled to the one or more channel adapters for translating channel program commands and determining, from the channel program, a target unit for which at least one channel program command is transmitted. In addition, a set of equipment controllers is provided which interpret channel program commands and status information, and which control data transfer to and from the storage facilities in accordance with the channel program command. Each equipment controller is provided with a device for calling a storage control manager. In addition, a device is also provided for establishing a unit thread by choosing an equipment controller from the set of equipment controllers. Each equipment controller is chosen as a function of a type of equipment that the channel requests as a target. The programmable storage controller also includes a device for passing channel program data to the unit thread and a device for executing the unit thread by executing the chosen equipment controller.

In accordance with another aspect of the present invention, the equipment controller is further provided with a mechanism for calling a cache manager. In addition, the equipment controller may be provided with a device for prioritizing input and output to and from the storage facilities. The one or more channel adapters may comprise one or more channel interface circuits, and the first interface may be provided with a channel interface controller. In this regard, the second interface may be provided with at least one of random access and sequential storage device channel adapters.

The system may include a random access and/or sequential storage device, wherein the random access device may comprise a removable direct access storage device, and the computer may comprise a mainframe computer.

In yet another aspect of the present invention, the programmable storage controller is further provided with a dispatcher for controlling the operation of the programmable storage controller. The channel interface controller is provided with an interrupt processor, responsive to a command received by the one or more channel programs, for interrupting the dispatcher, thus causing establishment of another unit thread, and a device for calling the execution of the another unit thread. The channel interface controller may also be provided with a device for controlling the one or more channel interface circuits and further a device for retrieving at least one of the channel programs.

In accordance with yet another aspect of the present invention, the programmable storage controller further comprises a device for signalling that data to be read from the storage facilities is compressed, and a device for decompressing data. The programmable storage controller may

also be provided with a device for signalling that data to be written should be compressed, and a device for compressing data. The device for compressing and the device for decompressing may be implemented by embedded code, or they may be implemented by the use of parallel RISC processors.

In accordance with yet a further aspect of the present invention, the second interface is provided with at least one standard high-speed parallel interface.

The computer may comprise a general purpose computer; the programmable storage controller may be provided with a device for accommodating a plurality of different kinds of storage devices which comprise the storage facilities; and the programmable storage controller may further be provided with a device for accepting and executing storage related commands from host channels having different program languages and dialects.

In accordance with a further aspect of the present invention, the computer of the storage controlling system may be configured in the form of a general purpose microcomputer, a personal computer, or a tightly coupled multi-processor.

In accordance with yet a further aspect of the present invention, the general purpose computer comprises a specialized operating system.

In yet a further aspect of the present invention, the second interface is further provided with an interface to another storage controller, wherein the programmable storage controller comprises a communication bridge for communicating with the another storage controller. The programmable storage controller may be provided with a plurality of customized controller services, such as a caching algorithm, and a device for performing data compression and decompression.

The storage controlling system may include the controller services. In addition, the storage controlling system may also include the at least one host and the storage facilities.

In a further aspect of the present invention, the at least one host may include at least one of a mainframe, a mini, and a microcomputer.

In a yet further aspect of the present invention, the equipment controllers comprise reentrant stored programs.

In an alternative embodiment of the present invention, a storage controlling method is provided for transferring information between one or more different types of target units and one or more channels of at least one host. The storage controlling method includes a number of steps such as receiving one or more channel programs transmitted from the one or more channels of the at least one host, wherein the channel program carries data, status, information and commands. During execution of the method, storage facilities, which include the one or more target units, are interfaced, and exchanges of storage data to and from the target units are controlled.

In a particular aspect of this alternative embodiment, channel program commands are translated in order to determine a target unit for which at least one channel program is transmitted, a unit thread is established, and channel program data is passed to the unit thread. The unit thread is established by choosing an equipment controlling routine from a set of equipment controlling routines which interpret data control commands and status information and control data transfers to and from storage facilities in accordance with the channel program command. Each equipment controlling routine is chosen as a function of the type of equipment that the channel program requests as a target. In addition, each equipment controlling routine, when executed, calls or initiates a storage control management routine.

During passing of channel program data to the unit thread, the unit thread is executed by executing the chosen equipment controlling routine.

The target units may comprise random access and sequential storage devices, and the type of equipment for which the host is configured may be different than the target units.

In a particular aspect of the alternative embodiment, the data being transmitted to and from the storage facilities is cached. In addition, the inputs and outputs to and from the storage facilities may be prioritized. In accordance with another particular aspect of the present invention, during execution of the equipment controlling routine, data is transferred to and from random access and sequential storage devices. The random access devices may include a removable direct access storage device while the computer comprises a mainframe computer.

In accordance with yet another aspect of the present invention, an interrupt processing is executed responsive to a command received by the one or more channel programs, thus causing another unit thread to be established and executed.

For purposes of clarification in defining the present invention in the following disclosure, the below-listed terms used herein are defined:

microprocessor	a central processing unit (CPU) on a single chip
computer	a machine that performs three functions: accepts structured input, processes it according to prescribed rules, and produces the results of the processing as output; examples of computers include super computers, mainframes, super mini computers, mini computers, work stations, personal computers, and microcomputers
general purpose computer	a computer which is provided with enough facilities to allow it to implement a wide range of different unrelated operating systems and/or applications
personal computer	a computer designed for use by one person at a time
microcomputer	a computer which runs with the use of a single-chip microprocessor; a microcomputer is typically less powerful than a mini computer and a mainframe computer
central processing unit	the computational and control unit of a computer; the CPU is the chip that functions as the "brain" of a computer
channel	a path or link through which information passes between two devices; a channel can be either internal or external to a computer
embedded code	code that is built into its carriers rather than associated with or called by them when needed; embedded code is used to make a program run faster or more efficiently or to provide a capability not available in a high-level language
RISC processor (reduced instructions set computing)	a type of microprocessor designed to efficiently process a relatively small set of instructions; the number of instructions built into the microprocessor is limited so that each instruction may be optimally carried out very rapidly, usually within a single cycle

-continued

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is further described in the detailed description which follows, by reference to the noted plurality of drawings, by way of non-limiting examples, of preferred embodiments of the present invention, in which like reference numerals represent similar parts throughout the several views of the drawings, and .wherein:

FIG. 1 illustrates basic elements of the storage controlling system of the present invention;

FIG. 2 illustrates a block diagram of the various elements which comprise the programmable storage controller of the present invention;

FIGS. 3*a*, 3*b*, 3*c* and 3*d* illustrate a number of flow charts which explain the general flow of the programmable storage controller according to a first embodiment of the present invention;

FIG. 4 illustrates an additional embodiment of the invention; and

FIG. 5 illustrates a third embodiment of the storage controlling system of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to FIG. 1, a storage controlling system 10 is illustrated which comprises a host system 12, a programmable storage controller 14, and storage facilities 16. Host system 12 is provided with one or more storage controller channels 13 which are connected to programmable storage controller 14 through a first interface 18. Storage facilities 16 are also connected to programmable storage controller 14, through a second interface 20.

Programmable storage controller 14 is implemented on a computer 15 having a first interface 18 and a second interface 20. Computer 15 is coupled to host system 12 via host connectors 22, which form a connection between storage controller channels 13 and first interface 18. Meanwhile, storage facilities 16 are connected to second interface 20 of the computer 15 via storage facility connectors 24.

Depending upon the type of computer which is used to implement the programmable storage controller 14, the type of host system being used, the type of storage facilities, and the host and storage facility connectors, different types of first and second interfaces 18, 20 may be implemented with various types of hardware. For example, specialized mainframe interface hardware may be constructed in order to provide the needed connections and signalling protocol to allow communication between host system 12 and computer 15. Second interface 20, which is coupled to storage facilities 16, may comprise one or more standard high-speed parallel interfaces, such as SCSI interfaces.

In a case where the host system is an IBM mainframe complex, host connectors 22 comprise bus and tag cables. Storage facility connectors 24 may be, e.g., configured in the form of ribbon cables, which extend to one or more drive mechanisms of storage facilities 16.

Depending upon the type of computer being used to implement the programmable storage controller 14, the type of host system, and the type of bus and tag cables which are connected to storage controller channels 13 of the host system, first interface 18 must have a number of features which render computer 15 compatible with the existing hardware of the host system and the bus and tag cables. For example, first interface 18 must have the following characteristics (in the case of an IBM ES 9000 mainframe host computer connected to an IBM compatible personal computer used to implement the programmable storage controller): First interface 18 (which will usually comprise an interface card) must conform to the specific signalling protocol (i.e., channel language) requirements of the host channel, such as Federal Information Processing Standard Publication for the Block Multiplexor I/O Channel Interface (FIPS-60); See FIPS PUB. 60-2, which is expressly incorporated by reference herein in its entirety. In addition, the interface card of first interface 18 must conform to the specifications for the bus architecture used to transfer data between interface 18 and the center processor and memory of computer 15.

Host system 12 requests data via an addressing scheme which assigns each block of data a device address, cylinder, and track. These address parameters are then translated by a disk controller into the actual physical address based on the controller's knowledge of the peripherals attached to it. The host operating system must receive consistent responses/results in accordance with the requests that it makes. For example, it expects the same data to be returned or written when it presents the same logical address. Programmable storage controller 14 and computer 15 are simply attached to the storage controller host system 12, channels of and may be configured to provide many general functions, such as: (1) emulating all of the responses of a standard storage controller which typically supports the host system; (2) translating each data address requested to physical addresses that can be satisfied by the storage facilities connected thereto; and (3) enhancing the performance of the storage facilities through use of, e.g., compression and decompression of data, disk striping, and caching.

In the embodiments of the storage controlling system which are discussed in the following description, programmable storage controller 14 is implemented by means of software code used to configure and run computer 15. At the same time, host system 12, storage controller channels 13, host connectors, storage facility connectors, first interface 18, second interface 20, and storage facility 16 are each implemented by means of hardware. Although these elements are disclosed as being implemented respectively with software and hardware, any one or all of the elements of the storage controlling system may be interchanged with hardware, firmware and/or software, depending on the type of host system, and design factors, such as efficiency of operation and cost.

Referring to FIG. 2, programmable storage controller 14 is provided with a channel adapter interface (CI) 210, an application interface (or dispatcher) 212, one or more controller emulators (equipment controllers) 214, a storage control manager 216, a cache manager 218, an I/O facilities handler (IOF) 220, a set of compression/decompression routines 222, and a device driver 224. Channel adapter interface 210, which is preferably implemented with software, provides a mechanism which allows application interface 212 to control other portions of the programmable storage controller, and also control channel adapter boards 26 (and thus the storage controller channels 13 of host system 12). CI 210 provides data, status, and commands from the channel, in the form of a channel program, to one or more controller emulators 214 which control data and status command exchanges with one or more target units. Application interface 212 handles command requests which are detected by CI 210 as incoming data streams. CI 210 is provided with an interrupt processor which causes execution thereof, and thus sets application interface 212 into action.

Application interface 212 routes the incoming commands (in the form of channel programs) to one or more controller emulators 214, by establishing and scheduling one or more unit threads based on the particular device (target unit) requested by the host system through the channel program. Application interface 212 uses a configuration table to determine which controller emulator should receive a request based upon the type of equipment the channel expects as a target. If the controller emulator which is needed to establish the unit thread is not currently loaded in memory of computer 15, application interface 212 will request the program to be loaded.

For each new target unit requested by a particular channel, application interface 212 initiates a new thread (by building a unit thread semaphore, having an assigned number corresponding to the requested target unit). Each unit thread comprises a controller emulator, (i.e., an equipment controller) which corresponds to the particular device (i.e., target unit) requested by the channel. Upon execution of the one or more controller emulators, the commands which are sent from the channel are retrieved by controller emulator 214 through use of channel adapter interface 210. Controller emulator 214 acts upon each valid command received by the host channels, by either reading or writing data, or performing some other miscellaneous processing, such as setting a file mask, configuring one or more particular devices, or issuing data and statuses of particular devices and data. Upon completion of one of these processes in accordance with the command received, a status is sent back to host system 12, signalling the state of the processing (e.g., completion of the command), and the results of the processing.

During execution of controller emulator 214, if the command received from the channel requires storage facility data to be referred, storage control manager 216 is called, which in turn calls cache manager 218 in order to establish access of the data requested to the controller emulator 214. If the requested data is not readily available in the cache, cache manager 218 then calls I/O facilities handler (IOF) 220 in order to execute a physical I/O of the data. If the data is compressed and must be decompressed, IOF 220 then calls compression/decompression routines 222 during retrieval of the data. Similarly, if IOF 220 is called to physically send data to the storage facilities, and if this data is to be compressed, compression/decompression routines 222 are again called. IOF 220 sends or retrieves data to or from storage facilities 16 via device driver 224.

A controller emulator is defined by the set of channel commands and channel programs which it interprets. The characteristics of a controller emulator are dependent on the definition of each channel command and channel program.

The controller emulators may interpret the channel commands and channel programs to control devices such as fixed and variable length record direct access storage devices, sequential access storage devices, and communication controllers.

The controller emulator must interpret each channel command and produce results consistent with the controller protocol which it is emulating. That is, the operation of the controller emulator should be transparent to the end user, the operator, and to the operating system hardware configuration parameters.

A unique and separate controller emulator is provided for each unique and separate target unit "type" specified by the channel. A unit thread is provided for each target unit specified by the host channel. Thus, since there may be many target units of the same type, there will typically be a plurality of unit threads assigned to a single controller emulator. Should the host be an IBM mainframe computer, which typically communicates with direct access magnetic storage using ECKD (extended count key data) format, the controller emulator must perform a somewhat more complicated translation or interpretation function in order to adapt the commands and data to an appropriate generic intermediate format readable by the device driver. On the other hand, if the host merely uses a fixed block I/O, the controller emulator will be much less complicated than the case for ECKD.

For specific examples of Direct and Sequential storage device channel command and program definitions which the controller emulators may be configured to interpret, please refer to *IBM* 3990 *Storage Control Reference* (GA32-0099-3) and *IBM* 3480 *Magnetic Tape Subsystem Reference: Channel Commands, Status and Sense Bytes, and Error Recovery Procedures* (GA32- 0042-6), each of which are expressly incorporated herein by reference in their entireties.

Host system 12 physically attaches to a storage controller through channel 13 which uses a unique addressing scheme. The programmable storage controller 14 may communicate with multiple channel interface cards of the channels, thus allowing multiple host channels to communicate with a single storage subsystem (i.e., the combined system of programmable storage controller 14 and storage facility 16). This allows multiple assembly host architectures (e.g., the IBM 370 Class, and Unisys 2200 Class) to communicate with a single parameter storage controller. The adapter card which is in the channel of the host system receives and passes on channel program commands and requests which are configured in a channel specific format. Meanwhile, application interface 212, which knows what type of host is connected to the channel adapter interface of the host, chooses an appropriate controller emulator which corresponds to the host, and assigns a unit thread representing that controller emulator.

The controller emulator 214 translates the channel program commands/requests from a channel-specific format to a "generic" format including generic address information and generic requests. Thereafter, storage control manager 216 receives the generic address and generic request information from the controller emulator 214, and translates the generic address/request to a physical target ID and data address which can then be routed to the IOF 220. The IOF 220 controls distribution of the request to the caching, striping, compression/decompression mechanisms, and/or device driver 224.

A particular embodiment of programmable storage controller 14 will now described with reference to FIGS. 3*a*–3*d*.

The application interface (or dispatcher) flow of operation is described in FIG. 3*b*. During execution of the application interface, in step S21, a packet is built into interface 18 for the CI which contains subchannel and command table information. Thereafter, in step S22, execution of the CI initialization (FIG. 3*a*, steps S1–S5) is called. Upon completion of CI initialization, which results in the execution of steps S1–S5 (FIG. 3*a*), the application interface then builds one or more unit thread semaphores; one unit thread semaphore is built for each device configured in the host system. Thereafter, in step S24, the one or more unit threads are scheduled, and thus initialized. Upon completion of scheduling/initialization of the unit threads, in step S25, a "get command" semaphore, is created. Then, steps S26–S28 are executed in response to clearance of the "get command" semaphore as indicated at step S26, which queues on the "get command" semaphore. Upon building an initial selection request packet for the CI in step S27, the application interface calls the CI for execution of the read channel processing (steps S6–S15, FIG. 3*a*).

An initial selection request packet, which is built in step S27, performs the functions of allowing initial device selection, storing target addresses and channel commands, providing initial status, and supporting data transfer functionality between the host channel and the programmable storage controller.

The CI (i.e., the channel adapter interface) functions as a device driver for carrying out I/O to and from the one or more storage controller channels 13 of host system 12. Thus, whenever data is either written to or read from the one or more storage controller channels 13, the CI must be executed in some fashion. In order to execute a "read channel" function, the read channel portion of the CI is called (see steps S6–S15). In order to execute a "write channel" function, the "write to" channel portion of the CI is called for execution (steps S16–S20).

FIG. 3*c* shows the flow of an individual controller emulator, which, when executed, comprises a unit thread. At step S30, the controller emulator waits for its unit thread semaphore to be cleared. Since there may be a plurality of unit threads which may be specified by the application interface (at step S23 thereof), each unit thread semaphore is assigned a particular number. Upon clearance of the unit thread semaphore (with the appropriate number), at step S31, the controller emulator retrieves the command and parameters from the CI. It is in this step (S31) that the controller emulator clears the appropriate semaphore, which allows reading of the data and commands by the CI during execution of the read channel processing. In addition, certain host-specific processing, such as translation and/or interpretation of the particular dialect of the I/O signaling protocol, if necessary, is performed in step S31. The controller emulator parses the command in step S33; that is, it is determined whether or not the command is valid. If the command is not valid, an error status is issued in step S39, and the processing returns to step S30 where the unit thread sleeps until its semaphore is again cleared.

If the command received by the CI is determined to be valid in step S33, another determination is made in step S34 as to whether or not the controller emulator must reference stored data for either a read or write operation. (it is noted that when the controller emulator is instructed to either read or write data to or from the storage facilities, the programmable storage controller must reference stored data in some way). If the controller emulator must reference stored data, the storage/cache manager is called in step S35. If the controller emulator is not required to reference stored data, one or more miscellaneous processings will be performed, depending upon the type of command received from the channel. This processing is indicated at step S40. Some examples of such miscellaneous processing include setting a file mask, configuring the devices, and issuing data and status information relating to the devices.

In step S35, the stored data which must be referenced is so referenced. That is, the storage/cache manager finds the data and returns an index to the controller emulator indicating where the controller emulator may find the data. Once this data is indexed in step S35, the controller emulator then makes another determination at step S36 as to what type of command is to be executed as instructed by the storage controller channel of the host system. If it is a read command, the controller emulator sends the data and status to the host by calling the CI (write to channel) (steps S16–S20), and returns to step S30 where it sleeps until the unit thread semaphore is again cleared. If the command by the host is a write command, the controller puts the data to be written in the cache buffer and sends a status back to the host by the use of the CI (write to channel) program. Upon sending this status, the controller emulator returns to step S30. After execution of each of steps S37, S38, S39, and/or S40, prior to returning to step S30, in step 40.1, the controller emulator clears the get command semaphore, thus allowing the application interface to execute steps S26–S38 and retrieve the appropriate command for use by the controller emulator.

As shown in FIG. 3a, the channel adapter interface (CI) comprises three processings including initialization, read to channel, and write to channel processings. The initialization processing comprises steps S1–S5. In step S1, the CI sets the get command semaphore. Upon setting of the get command semaphore, the processing then loads a command table in step S2, loads a subchannel table in step S3, and initializes the interface card of the storage controller channel 13 in step S4. Thereafter, in step S5, the channel interface returns to where it was called, which in this case is the application interface at step S22.

The read channel processing of the CI comprises steps S6–S15. In step S6, the interrupts of the storage controller channels are disabled. Upon disabling of the interrupts of the storage controller channel interface card, in step S7, the CI sets the initial selection function, which in turn allows the host to perform initial device selection. Once the initial selection function is set, the CI waits for a device (target unit) selection by the storage controller channel in step S8. Then, in step S9, the CI puts the address and command of the device (target unit) selection in the initial selection request packet, which is built into step S27 of the application interface (FIG. 3b). Thereafter, in step S10, the unit thread semaphore corresponding to a desired controller emulator is cleared, thus causing the appropriate controller emulator to be executed. In step S11, the CI waits for the read data semaphore to be cleared. This occurs in step S31 of the controller emulator, where the controller emulator gets the command and parameters from the CI. If an I/O is necessary, in step S12, the CI performs such channel I/O. Thereafter, in step S13, the data requested by the host channel (subsystem data) is sent to the host channel. The read complete semaphore is then cleared in step S14, and the processing is returned to where it was called, which is, in this case, step S28 of the application interface.

The write to channel processing of the CI comprises steps S16–S20. In step S16, the write to channel processing of the CI sets the write request semaphore, thus prohibiting or preventing any unwanted or unauthorized concurrent write to channel requests by other controller emulators of the programmable storage controller. Thus, the write request semaphore ensures that the status which is being transferred to the channel by use of the write to channel processing is not disturbed. Once the write request semaphore is set, in step S17, the status is sent from the controller emulator to the interface card of the controller channel. Once the status is sent, the write request semaphore is cleared in step S19, thus freeing use of the write to channel processing of the CI by another controller emulator which may be waiting in line. At step S20, the processing is returned to where it was called, which, in this case, is at either of steps S37 and S38 of the controller emulator.

As shown in FIG. 3d, the storage/cache manager comprises two main processings: a main activity, and a background activity. The main activity is that which is activated when the storage/cache manager is calling by the controller emulator. For example, when called in step S35 of the controller emulator (see FIG. 3c), the main activity of the storage/cache manager is activated, which causes steps S41–S46 (see FIG. 3d) to be executed. At step S41, a determination is made as to whether the requested data is in cache. If the requested data is in cache, the processing skips to the return step, S46, and the storage/cache manager returns an index indicating where the requested data is in the cache, thus allowing the controller emulator to access the data. If, however, the requested data is not in cache, as determined in step S31, another determination is made in step S42 as to whether one or more necessary cache buffers are available. If not, an existing buffer is reallocated in step S43. However, if the cache buffer is available, the processing immediately proceeds to step S44 where an I/O request packet is built for a read operation. Thereafter, the IOF is called in step S45, which results in a physical I/O of data between the cache buffer and a storage device. In step S46, the processing returns to the controller emulator.

The background activity of the storage/cache manager is performed by steps S47–S53. In step S47, the manager sets the unit thread priority of the system. Then, in step S48, the storage/cache manager waits until the system becomes idle. Once there is idle time in the system, the storage/cache manager will search for an updated record. If the updated record is found in the determination step S50, the processing proceeds to step S51, where an I/O facilities packet for writing is built. The IOF is called in step S52, and the cache tables are updated in step S53. Thereafter, the system returns to step S48, where the storage/cache manager either waits for the main activity processing to be called, or for the system to become idle and for an updated record, in steps S48 and S49.

IOF 220 interfaces with storage facilities via a device driver 224 (see FIG. 2). One or more device drivers may be provided depending on the particular peripheral storage devices which are connected to second interface 20 of the storage controlling system 10 (see FIG. 1). In a preferred embodiment, a device driver is provided as disclosed in copending U.S. patent application Ser. No. 07/882,003, filed concurrently with the present application, now abandoned, the disclosure of which is expressly incorporated herein in its entirety. The device driver disclosed therein is capable of self-configuring itself upon IPL (Initial Program Load) to support a number of different peripheral devices for input and output thereto.

In calling of either of the channel adapter interface 210, and the device driver 224 an IOTCL interface is used to avoid the overhead of the host controller operating system file structure system checks. This allows more efficient access directly to the media of the storage facilities.

Should the programmable storage controller be provided with disk striping, it is noted that IOF 220 (FIG. 2) should include a mechanism for executing one or more switch lists and priority ranked service queues, prioritizing I/O, and routing to appropriate physical storage devices and channels. It is noted that a number of separate and independent reads and writes will be simultaneously executed for certain portions of data if data striping is utilized.

With respect to the embodiment previously discussed, there are a number of features or modifications which may provide this embodiment with certain advantages. For example, the compression/decompression mechanisms may be implemented by use of dynamic data compression algorithms. The compression/decompression mechanisms may utilize either static or adaptive techniques. Examples of static compression/decompression techniques include, but are not limited to, run-length encoding, bit mapping (Huffman), and arithmetic coding. Adaptive techniques may include, for example, Huffman and the Ziv-Limpel (LZ) variants. Data compression/decompression mechanisms are described in some detail in *Text Compression*, Timothy C. Bell, John G. Cleary, Ian H. Witten, 1990 Prentis Hall, Inc., which is expressly incorporated herein by reference in its entirety.

In addition, ASIC's (application specific integrated circuits) and high density gate arrays may be used to implement various portions of the storage controlling system, such as the IOF, the device driver, and the second interface, thus enhancing I/O. Microprocessors such as the Intel 80486 and 80586, or the Motorola 68030 and 68040, may be used to implement computer 15. These microprocessors provide great amounts of compute power in very small space, inexpensively, and within a relatively hostile environment, e.g., under undesirable environmental conditions such as heat and humidity.

In one modification to the first embodiment of the present invention, the programmable storage controller is implemented with a specialized operating system in conjunction with a tightly coupled multi-processor. The specialized operating system should be provided with a number of features including tightly coupled multi-processor support, a sophisticated multi-tasking scheduler having preemptive support and real time support, and memory allocation. In addition, the specialized operating system should not have too many features; that is, the OS should be provided with only the "bare essential elements," without any significant video, keyboard, or character I/O support. It is also preferred that the specialized operating system have the ability to subdivide and swap operating system functions to allow an application program and only the essential OS services to use the processors at one time.

A number of new peripheral technologies can be considered as particularly beneficial and desired for the storage facilities connected to second interface 20 of storage controlling system 10. For example, rewritable optical disks, optical tapes, and 4 mm digital tapes (DAT) have a number of beneficial characteristics such as large storage capacities, quick access time, and low floor space and power requirements.

FIG. 4 illustrates a block diagram of a further embodiment of the present invention. In this embodiment, host system 12 is provided with first and second host computers 28, 30, and an additional "other" storage controller 32 is connected to interface 20 of computer 15. In order to effect communication between programmable storage controller 14 and other storage controller 32, a mechanism must be provided for interfacing other storage controller 32 with appropriate portions of programmable storage controller 14, so as to allow communication of various commands, status and data therebetween. This mechanism may include an inter-controller communication bridge. The inter-controller communication bridge may comprise of a separate controller emulator which provides translation functions allowing the host computer to communicate with storage controller 32. The controller emulator makes the appropriate translation of host requests to the generic format, which is then interpreted by the storage control manager. The storage control manager then further translates the instructions and commands into an appropriate protocol which may be interpreted by the other storage controller 32 as actions or commands to respond to.

This additional inter-controller communication bridge may be provided in order to accommodate functions such as backing up one storage channel with another, or to provide alternate routes of communication in case a channel interface or storage controller has been disabled because of hardware or software failure. It is noted that although FIG. 4 indicates a particular configuration of the inter-controller connection, other storage controller 32 may be connected to first interface 18 rather than second interface 20.

In order for the programmable storage controller 14 to accommodate a number of different host computers, such as first host computer 28 and second host computer 30, each unit thread which is established and scheduled by application interface 212 (see FIG. 2) should further be provided with an interpretor which may interpret the different channel languages and/or channel dialects of the communication interface used by the one or more host computers.

A channel language is defined to be the electrical and control signalling sequences of the host interface. A channel program dialect is defined to be the ordering and context sensitive nature of the command signals within a specific channel language. Examples of channel languages include FIPS 60 and SCSI (Small Computer Systems Interface). Examples of dialects of, e.g., the FIPS 60 channel language include FIPS 60 as implemented by Unisys and FIPS 60 as implemented by IBM.

Different types of host computers may be accommodated by use of an interpretor provided in the unit thread, each interpretor being specific to each particular host computer, while each controller emulator is chosen depending on the target unit specified by the channel of the host computer. The interpretor not only interprets the different commands sent by the channel of the host computer, but also interprets (for the host) the return status information being sent back to the host computer.

Another embodiment of the storage controlling system of the present invention is depicted in FIG. 5. As shown in this figure, one or more mainframe computers, which may be, for example, IBM mainframes, are connected to mainframe interface hardware 52 of a microcomputer 54. Microcomputer 54 includes an IBM compatible personal computer with, for example, either an 80486 or 80586 microprocessor. Microcomputer 54 is configured with a programmable storage controller 14 having a SCSI device driver 56, which interfaces with a plurality of SCSI interfaces connected to optical disks 60, which comprise the storage facilities in the present embodiment.

This second embodiment is explained below with reference to an IBM 3990 mainframe disk controller; however, it is easily applicable to any IBM mainframe computer or minicomputer which uses ECKD to send and retrieve data to and from magnetic disks. The system shown in FIG. 5

provides a full scale emulation of the IBM 3990 mainframe disk controller and its associated magnetic disk using a microcomputer and one or more rewriteable optical disks 60. The programmable storage controller 14' may be provided with compression and decompression algorithms. In addition, mechanisms may be provided which allow striping of I/O operations so that each I/O to and from optical disk 60 is subdivided into several parallel read and write efforts. In addition, the microcomputer may be provided with mechanisms for caching, and a staging device for read operations and look ahead reads.

The embodiment of FIG. 5 has a number of benefits in that magnetic disks which are supported by mainframes have a relatively large footprint, and are extremely expensive to support. On the other hand, rewritable optical disks, as do other optical storage media, allow data storage at a fraction of the cost, space, and environmental requirements of the magnetic disk.

While the invention has been described with reference to a number of preferred embodiments, it is understood that the words which have been used herein are words of description, rather than words of limitation. Changes may be made, within the purview of the appended claims, without departing from the scope and spirit of the invention in its various aspects. Although the invention has been described herein with reference to particular means, materials and embodiments, it is understood that the invention is not be limited to the particulars disclosed herein, and that the invention extends to all equivalent structures, methods and uses such as are within the scope of the appended claims.

What is claimed is:

1. A storage control subsystem connected between one or more storage controller channels of at least one host system and data storage facilities comprising a plurality of target units, said storage control subsystem comprising:

a programmable storage controller that emulates a plurality of types of target unit specific storage controllers, said programmable storage controller being implemented with an application program and a computer, said computer being configured by said application program;

a first interface for interfacing a plurality of channel adapters which carry a plurality of channel programs transmitted from the channels of the host system to said programmable storage controller, each channel program having means for carrying data, status information and commands; and

a second interface for interfacing said programmable storage controller to said target units;

said programmable storage controller comprising a plurality of controller emulators, said controller emulators comprising means for translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller that includes generic address and request information, to thereby facilitate data and status command exchanges with said plurality of target units.

2. The storage control subsystem according to claim 1, wherein said programmable storage controller comprises:

means coupled to each said channel adapter for translating channel program commands and determining, from each channel program, a target unit for which at least one channel program command is transmitted;

a set of equipment controllers which interpret channel program commands and status information, and control data transfer to and from said storage facilities in

accordance with the channel program commands, each equipment controller comprising means for calling a storage control manager;

means for establishing a unit thread by selecting an equipment controller from said set of controllers, each equipment controller being selected by said means for establishing as a function of the type of equipment that the channel program requests as a target;

means for passing channel program data to said unit thread; and

means for executing said unit thread by executing said equipment controller.

3. The storage control subsystem according to claim 2, wherein said storage control manager comprises a cache manager.

4. The storage control subsystem according to claim 2, wherein said equipment controller further comprises means for prioritizing input and output to and from said storage facilities.

5. The storage control subsystem according to claim 2, wherein each of said channel adapters comprises one or more channel interface circuits, and further wherein said first interface comprises a channel interface controller.

6. The storage control subsystem according to claim 2, wherein the type of equipment for which the host is configured is different than one of said target units.

7. The storage control subsystem according to claim 5, wherein said second interface comprises at least one of a random access and a sequential storage device channel adapter.

8. The storage control subsystem according to claim 7, wherein said storage control subsystem comprises at least one of a random access and a sequential storage device.

9. The storage control subsystem according to claim 8, wherein said storage control subsystem comprises a random access device which comprises a removable direct access storage device.

10. The storage control subsystem according to claim 9, wherein said host system comprises a mainframe computer.

11. The storage control subsystem according to claim 5, wherein said programmable storage controller further comprises a dispatcher for controlling the operation of said programmable storage controller, and further wherein said channel interface controller comprises an interrupt processor, responsive to a command received by each of said channel programs, for interrupting said dispatcher, thus causing said means for executing to execute another unit thread.

12. The storage control subsystem according to claim 11, wherein said channel interface controller further comprises means for controlling said one or more channel interface circuits and means for retrieving at least one of said channel programs.

13. The storage control subsystem according to claim 1, wherein said programmable storage controller further comprises means for signalling that data to be read is compressed, and means for decompressing data.

14. The storage control subsystem according to claim 13, wherein said programmable storage controller further comprises means for signalling that data to be written should be compressed, and means for compressing data.

15. The storage control subsystem according to claim 14, wherein said means for compressing data and said means for decompressing data comprise embedded code.

16. The storage control subsystem according to claim 14, wherein said means for compressing and said means for decompressing comprise parallel RISC processors.

17. The storage control subsystem according to claim 1, wherein said second interface comprises at least one standard high-speed parallel interface.

18. The storage control subsystem according to claim 1, wherein said programmable storage controller further comprises storage control means for receiving and translating the generic address and request information to physical target ID and data address information.

19. The storage control subsystem according to claim 17, wherein said programmable storage controller comprises means for controlling a plurality of different kinds of storage devices which comprise said storage facilities.

20. The storage control subsystem according to claim 1, wherein said programmable storage controller comprises means for accepting and executing storage related commands from host channels having a plurality of different channel languages and dialects.

21. The storage control subsystem according to claim 1, wherein said computer comprises a microcomputer.

22. The storage control subsystem according to claim 1, wherein said computer comprises a personal computer.

23. The storage control subsystem according to claim 1, wherein said computer comprises a tightly coupled multiprocessor.

24. The storage control subsystem according to claim 1, wherein said computer comprises a specialized operating system.

25. The storage control subsystem according to claim 1, wherein one of said first and second interfaces further comprises an interface to another storage controller, and further wherein the programmable storage controller comprises a communication bridge for communicating with said another storage controller.

26. The storage control subsystem according to claim 1, wherein said programmable storage controller comprises means for providing a plurality of customized controller services.

27. The storage control subsystem according to claim 26, wherein said means for providing controller services comprise at least one of: (i) a caching algorithm; and (ii) means for performing data compression and decompression.

28. The storage control subsystem according to claim 26, wherein said storage control subsystem further comprises said means for providing controller services.

29. The storage control subsystem according to claim 1, wherein said storage control subsystem further comprises said at least one host and said storage facilities.

30. The storage control subsystem according to claim 1, wherein said at least one host system comprises at least one of a mainframe computer, a mini computer, and a micro computer.

31. The storage control subsystem according to claim 2, wherein said equipment controllers comprise one or more reentrant stored programs.

32. The storage control subsystem according to claim 1, wherein said computer comprises a general purpose computer having an operating system comprising means for controlling said general purpose computer.

33. The storage control subsystem according to claim 32, wherein said operating system comprises a standard operating system.

34. A storage controlling method for transferring data between one or more channels of at least one host system and data storage facilities comprising a plurality of different types of target units, said method comprising:

operating a programmable storage controller that emulates a plurality of types of target unit specific storage

controllers, said programmable storage controller being implemented with an application program and a computer, said computer being configured by said application program;

receiving a plurality of channel programs transmitted from the channels of the host system, to said programmable storage controller, each channel program carrying at least one of data, status information and commands; and

interfacing said programmable storage controller with said target units; and

controlling, with said programmable storage controller, exchanges of storage data to and from said target units, said controlling comprising translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller, said generic format including generic address and request information.

35. The storage controlling method according to claim 34, further comprising prioritizing input and output to and from said storage facilities.

36. The storage controlling method according to claim 34, wherein the method further comprises caching data being transmitted to and from said storage facilities.

37. The storage controlling method according to claim 34, wherein interrupt processing is executed responsive to a command received from the at least one host system, thus causing another unit thread to be executed.

38. The storage controlling method according to claim 34, wherein said one or more different types of target units comprise random access and sequential storage devices.

39. The storage controlling method according to claim 34, said method further comprising:

translating channel program commands, and determining from the channel program a target unit for which at least one channel program command is transmitted;

establishing a unit thread by choosing an equipment controlling routine from a set of equipment controlling routines which interpret data control commands and status information and control data transfer to and from storage facilities in accordance with the channel program commands, each equipment controlling routine being chosen as a function of the type of equipment that the channel program requests as a target;

passing channel program data to the unit thread; and executing the unit thread by executing the chosen equipment controlling routine; and

said method, during execution of said equipment controlling routine, performing storage control management.

40. The storage controlling method according to claim 39, wherein, during execution of said equipment controlling routine, data is transferred to and from at least one of random access and sequential storage devices.

41. The storage controlling method according to claim 40, wherein the data is transferred to and from a random access storage device which comprises a removable direct access storage device, and further wherein the at least one host system comprises a mainframe computer.

42. The storage controlling method according to claim 39, wherein the type of equipment for which the host system is configured is different than said one or more different types of target units.

43. The storage controlling method according to claim 34, wherein said controlling further comprises receiving and translating the generic address and request information to physical target ID and data address information.

44. The storage controlling method according to claim 34, wherein said computer comprises a general purpose computer having an operating system, said operating comprising controlling said general purpose computer with said operating system.

45. A storage control subsystem connected between storage controller channels of at least one host system and data storage facilities comprising a plurality of target units, said storage control subsystem comprising:

a programmable storage controller that emulates a plurality of types of target unit specific storage controllers, said programmable storage controller being implemented with a general purpose computer with a general purpose operating system supporting an application program, said general purpose computer being configured by said application program;

a first interface for interfacing a plurality of channel adapters which carry a plurality of channel programs transmitted from the channels of the host system to said programmable storage controller, each channel program having means for carrying data, status information and commands; and

a second interface for interfacing said programmable storage controller to said target units;

said programmable storage controller comprising a plurality of controller emulators, said controller emulators comprising means for translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller that includes generic address and request information, to thereby facilitate data and status command exchanges with said plurality of target units.

46. The storage control subsystem according to claim 45, wherein said programmable storage controller further comprises storage control means for receiving and translating the generic address and request information to physical target ID and data address information.

47. A storage controlling method for transferring data between one or more channels of at least one host system and data storage facilities comprising a plurality of different types of target units, said method comprising:

operating a programmable storage controller that emulates a plurality of types of target unit specific storage controllers, said programmable storage controller being implemented with a general purpose computer with a general purpose operating system supporting an application program, said general purpose computer being configured by said application program;

receiving a plurality of channel programs transmitted from the channels of the host system, to said programmable storage controller, each channel program carrying at least one of data, status information and commands; and

interfacing said programmable storage controller with said target units; and

controlling, with said programmable storage controller, exchanges of storage data to and from said target units, said controlling comprising translating said channel programs and commands from a channel specific format to a generic format of said programmable storage controller, said generic format including generic address and request information.

48. The storage controlling method according to claim 47, wherein said controlling further comprises receiving and translating the generic address and request information to physical target ID and data address information.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.	:	5,530,845
DATED	:	June 25, 1996
INVENTOR(S)	:	D. HIATT et al.

It is certified that error appears in the above-indentified patent and that said Letters Patent is hereby corrected as shown below:

On the title page, item: [56], "References Cited", "U.S. PATENT DOCUMENTS"; add —5,101,490 3/1992 Getson, Jr. et al.—, —4,479,179 10/1984 Dinwiddie, Jr.—, —4,987,530 1/1991 Wagner et al.—, —4,825,406 4/1989 Bean et al.—, and —4,896,262 1/1990 Wayama et al.—.

Signed and Sealed this

Fifteenth Day of July, 1997

Attest:

Bruce Lehman

BRUCE LEHMAN

Attesting Officer *Commissioner of Patents and Trademarks*

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 5,530,845 Page 1 of 1
DATED : June 25, 1996
INVENTOR(S) : David M. Hiatt et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page,
Item [75], Inventors,
Line 1, "Klos," should be changed to -- **Klos; Larry T. Jost** --.
Line 2, "both" should be changed to -- all --.

Signed and Sealed this

Sixteenth Day of April, 2002

Attest:

JAMES E. ROGAN
Director of the United States Patent and Trademark Office

Attesting Officer

B

Distributed Storage Networks

Architecture, Protocols and Management

Thomas C. Jepsen
IT Consultant; Programming Languages Editor,
IEEE ITPro Magazine,
Computer Science Instructor,
North Carolina State University, USA

John Wiley & Sons, Ltd

Copyright © 2003 John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,
West Sussex PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-470-85020-5

Typeset in 11/13pt Palatino by Laserwords Private Limited, Chennai, India
Printed and bound in Great Britain by Biddles Ltd, Guildford and King's Lynn
This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

A refinement on the arbitrated loop topology is provided by adding a *hub* to the configuration (see Figure 1.6). In a true arbitrated loop topology, failure of a single node will cause the entire loop to become inoperative, since the entire bandwidth passes through each device. The hub contains circuitry to sense device failure, and permit data intended for other nodes to pass through if a node becomes inoperative. The addition of a hub in the middle of the configuration changes the physical topology from a loop to a star, while retaining the logical loop configuration. Use of a hub provides better cable manageability and increased reliability. Devices may be added or removed from the loop while it is operating if a hub is used.

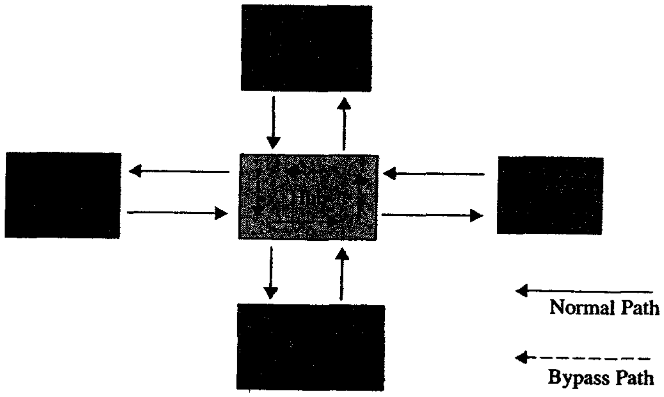


Figure 1.6 Arbitrated loop topology – hub configuration.

1.3.2.3 Switched Fabric Topology

A SAN utilizing switched fabric topology consists of computing nodes and storage nodes interconnected by means of a storage fabric (see Figure 1.7). A *fabric switch* enables any storage device to be connected to any computing device for the duration of a data transfer operation. A fabric switch can support multiple simultaneous full-bandwidth connections between storage and computing nodes. A *storage director* is a specialized type of fabric switch which provides enhanced management and reliability features, such as duplicated switch fabrics and power supplies. Use of the switched fabric also enables a common backup server to be connected to any of the storage devices for scheduled or manual backup purposes. A *gateway* may be employed to provide SAN/WAN interworking and protocol translation in distributed SAN applications.

C

storage area network

CONTEXT [Fibre Channel] [Network] [Storage System]

1. A network whose primary purpose is the transfer of data between computer systems and storage elements and among storage elements. Abbreviated SAN. A SAN consists of a communication infrastructure, which provides physical connections, and a management layer, which organizes the connections, storage elements, and computer systems so that data transfer is secure and robust. The term SAN is usually (but not necessarily) identified with block I/O services rather than file access services.
2. A storage system consisting of storage elements, storage devices, computer systems, and/or appliances, plus all control software, communicating over a network.

Note: The SNIA definition specifically does not identify the term SAN with Fibre Channel technology. When the term SAN is used in connection with Fibre Channel technology, use of a qualified phrase such as "Fibre Channel SAN" is encouraged. According to this definition an Ethernet-based network whose primary purpose is to provide access to storage elements would be considered a SAN. SANs are sometimes also used for system interconnection in clusters.

D

RAID

CONTEXT [Storage System]

1. An Acronym for *Redundant Array of Independent Disks*, a family of techniques for managing multiple disks to deliver desirable cost, data availability, and performance characteristics to host environments.
2. A Redundant Array of Independent Disks.
3. A phrase adopted from the 1988 SIGMOD paper *A Case for Redundant Arrays of Inexpensive Disks*.

E

Storage Networks: The Complete Reference

Robert Spalding

McGraw-Hill/Osborne
New York Chicago San Francisco
Lisbon London Madrid Mexico City
Milan New Delhi San Juan
Seoul Singapore Sydney Toronto

McGraw-Hill/Osborne
2600 Tenth Street
Berkeley, California 94710
U.S.A.

To arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact McGraw-Hill/Osborne at the preceding address. For information on translations or book distributors outside the U.S.A., please see the International Contact Information page immediately following the index of this book.

Storage Networks: The Complete Reference

1234567890 CUS CUS 019876543

ISBN 0-07-222476-2

Publisher
Brandon A. Nordin

Vice President & Associate Publisher
Scott Rogers

Acquisitions Editor
Franny Kelly

Project Editor
Jody McKenzie

Technical Editor
Giota Vavasis Then (Patty)

Copy Editors
Mike McGee
Lisa Theobald

Proofreader
Susie Elkind

Indexer
Karin Arrigoni

Computer Designers
Lucie Ericksen
John Patrus

Illustrators
Lyssa Wald
Michael Mueller
Alex Putney

Series Design
Peter F. Hancik

This book was composed with Corel VENTURA™ Publisher.

Information has been obtained by McGraw-Hill/Osborne from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, McGraw-Hill/Osborne, or others, McGraw-Hill/Osborne does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from the use of such information.

The Controller

The controller allows multiple devices to be connected to the adapter. As the adapter forms the primary path to and from the server, the controller provides the connectivity to the actual devices that communicate with the server. As indicated in Figure 6-6, the controller functions can be manifold but primarily control the flow of data to and from the devices. In doing so, it provides an addressing scheme for the devices, shields the server from the operational complexities of the device, and provides another temporary location for data (prior to it being written or cached) for faster access.

Like adapters, controllers consist of both hardware and software components. However, their primary functionality is driven by hardware microcode, as indicated in Figure 6-6.

By keeping track of a disk's position and being the initiator of the drive's read/write electronics (see the next section for more details on magnetic head disk assemblies, referred to as *HDA* mechanisms), the controller provides data buffering between the disk and the host interface or adapter with its error detection, correction, retry, and recovery functions. Both of these functions are susceptible to changes in connectivity due to their communications with the operating system.

The controller also offers the system's attached disks as an acceptable storage location even though the actual physical mapping may be very different. This allows the controller to translate I/O system requests into actual physical commands that can be executed on the real disk drive. Typical in most systems, the controllers and operating system view the disk drive locations as a linear pool of logical locations. This level of abstraction translation, though having existed in I/O configurations for many years, is necessary to translate logical blocks of data into physical addresses on the storage media.

This is a key concept within the controller functionality. It demonstrates how the operating system communicates within a logical set of storage locations that are considered virtual to the controller, which translates those locations into the physical realities the controller must manage. Figure 6-7 represents an overview of how logical views are mapped into physical views.

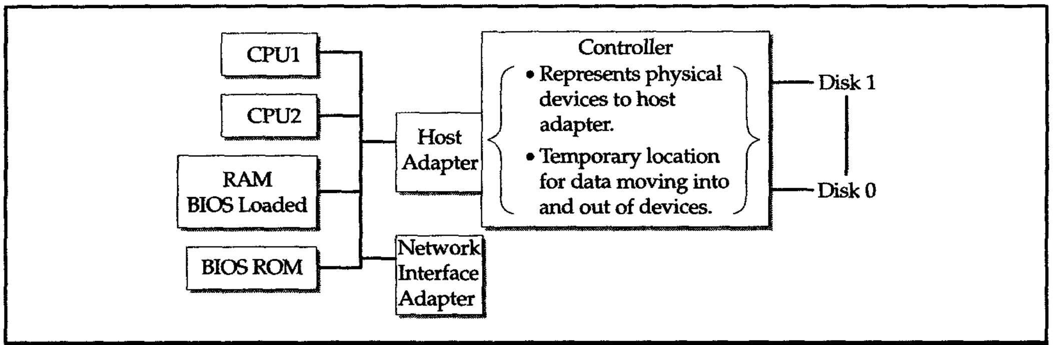

CPU1
CPU2
RAM
BIOS Loaded
BIOS ROM
Host Adapter
Network Interface Adapter
Controller
• Represents physical devices to host adapter.
• Temporary location for data moving into and out of devices.
Disk 1
Disk 0

Figure 6-6. Controller functions

single loop failure. In other words, should one loop go down, the information on the storage device can be retrieved via the second loop. A dual loop requires four of the switch's ports. Another, less typical method of attaching a JBOD array to a switch is to split the devices into separate loops. A JBOD array of eight drives could have one loop serving drives 1–4 and a second loop for drives 5–8. This method also requires four ports on the switch. The benefits of splitting the devices into several loops include shorter path lengths and less arbitration overhead within the loop itself.

The disadvantage of any JBOD implementation is its lack of fault resiliency, though there are software RAID products that allow the stripping of data with recovery mechanisms encompassed in the JBOD enclosure. Given the number of disks and the transmission of SCSI commands to multiple targets, using RAID software in conjunction with a JBOD implementation presents its own problems. It is important to keep in mind that while these are Fibre Channel-enabled disks, the disk drives themselves execute a SCSI command set when performing read/writes.

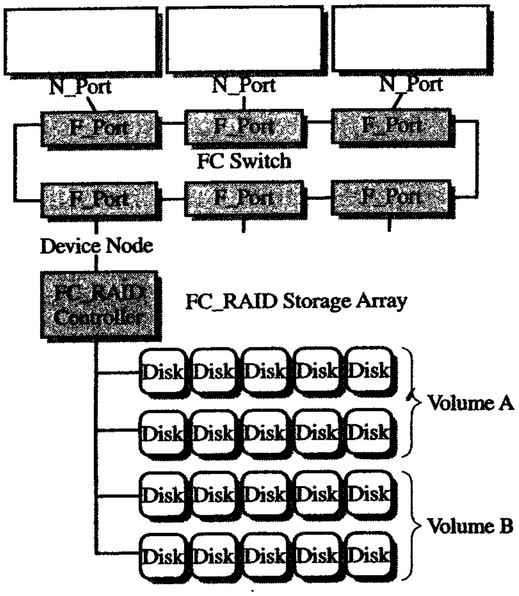
RAID

A Fibre Channel-enabled RAID storage array places a controller in front of the disk array that provides and controls the RAID level for the disk array (RAID 1–5). RAID offers a way of protecting the data by creating an array of disk drives that are viewed as one logical volume. Inside this logical volume, which may consist of seven drives, data is partitioned, as is recovery information. In the event of a single drive failure, the array reassembles the information on the remaining drives and continues to run. The RAID controller uses either an N_Port or a NL_Port depending on how the vendor put the disk enclosure together.

Because the RAID controller stands in front of the array, the FC-enabled disks, regardless of their configuration, become transparent. The switch only sees one device, not several linked together. A single RAID controller can also control several arrays, as indicated in Figure 14-7. For example, four volumes, each volume containing seven disks, would equal a RAID system of 28 disks. Still, the switch only sees the RAID controller—just one device. In this scenario, more often than not, the RAID controller will utilize an N_Port, not a loop.

The key advantage of Fibre Channel RAID is its ability to provide levels of fault resistance for hardware failures, a feature not found in JBOD configurations. For enterprise-level workloads, this feature is all but mandatory.

Just as the HBA is the critical point between a server's operating system and the switch's operating system, RAID controllers require a level of specific Fibre Channel software that must be compatible with the switch and the HBA. As noted previously, it is the HBA's job to inform the server which disks are available. In a JBOD configuration, this is pretty straightforward. Each disk is an addressable unit. In a RAID configuration, it becomes the controller's duty to specify which disk is addressable. The RAID controller, via the software contained within it, has to identify itself to the switch, specifically the Name Server within the switch, as well as the HBA,



order for the server to know which disks it has access to on the network. This can uickly become confusing, given that RAID deals in logical units, not independent ddressable disks.

One last word about Fibre Channel storage, and this goes for RAID and JBOD onfigurations: when assigning multiple servers to the switch (via the HBA), the servers have to be told which storage resources they are allowed to play with. And his can quickly become tedious. For example, each server has its own file system, and hat file system must reflect the location of the files the server has access to. Problems arise when two or more servers have access to the same files. What happens when two servers reach out for the same file? You guessed it… trouble, headaches, and a whole lot of shouting. File sharing between servers attached to a Storage Area Network remains a tedious and problematic issue. Consequently, zoning and masking each server's authorized resources continues to be a prerequisite for effective operation. Before *you* start shouting, and reach for the aspirin, have a look at Chapter 22.

dges and Routers

In addition to the challenges posed by file sharing, data sharing, and device sharing, there are standard data center practices that are required for any type of storage model. Every data center must protect the data stored within the arrays. Most accomplish this using backup/recovery software and practices, which entails the use of tape devices as

F

Designing Storage Area Networks

Second Edition

A Practical Reference for Implementing Fibre Channel and IP SANs

Tom Clark

Addison-Wesley
Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid • Capetown
Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales
(317) 581-3793
international@pearsontechgroup.com

Visit Addison-Wesley on the Web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

Clark, Tom, 1947 Aug. 12–
Designing storage area networks : a practical reference for implementing storage area networks / Tom Clark—2nd ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-321-13650-0 (alk. paper)
1. Computer networks. 2. Information storage and retrieval systems. 3. Computer storage devices. 4. Internetworking (Telecommunication) 5. Fibre Channel (Standard) I. Title.

TK5105.5.C547 2003
004.6—dc21 2003041450

For information on obtaining permission for use of material from this work, please submit a written request to:

Pearson Education, Inc.
Rights and Contracts Department
75 Arlington Street, Suite 300
Boston, MA 02116
Fax: (617) 848-7047

ISBN 0-321-13650-0
Text printed on recycled and acid-free paper.

ISBN 0321136500
2 3 4 5 6 7 CRS 06 05 04 03
2nd Printing July 2003

vendor-independent services for device discovery and status reporting. This enables management frameworks to solicit common information in a multivendor HBA environment, including traffic statistics and configuration data useful for monitoring the operation of the SAN. Additional information on the Common HBA API is available on the SNIA Web site at www.snia.org.

5.3 Fibre Channel RAID

Redundant arrays of independent disks, or RAIDs, are a well-established technology in traditional storage applications. RAID standards define methods for storing data to multiple disks and imply intelligence in the form of a RAID controller. The controller can be implemented in software—for example, as an application running on a file server—but typically is a dedicated card installed in a RAID storage enclosure. RAID storage configurations may include eight to ten disks for a departmental-level array, or many more for enterprise applications.

When a server is connected to a single disk drive, reads or writes of multiple data blocks are limited by the buffering capability and rotation speed of the disk. While the disk is busy processing one or more blocks, the host must wait for acknowledgment before sending or receiving more data. You can increase throughput by dispersing data blocks across several disks in a RAID, a technique called **striping**. In a write operation, for example, the host can send fewer data blocks to multiple targets consecutively and thereby avoid swamping the capacity of any individual drive. This simplified RAID is called level 0. Although it boosts performance, it does not provide data security. If a single disk fails, data cannot be reconstructed from the survivors.

Other RAID levels introduce data integrity by either writing parity data to a dedicated drive or writing parity information on each drive in an array. RAID level 3 writes byte-level parity to a dedicated drive. RAID level 4 writes block-level parity to a parity drive. Because a dedicated drive contains the information required to reconstruct data, levels 3 and 4 pose a data integrity problem if one of the data drives and the parity drive itself fail. RAID level 5 addresses this problem by striping block-level parity information across each drive. If an individual drive fails, it can be reconstructed from the parity information contained on the other drives. Another technique, level 1, achieves full data security by sacrificing the performance gain of striping in favor of simple disk mirroring. In **disk mirroring**, every write operation to a primary disk is duplicated on a secondary, or mirrored, disk. If the primary disk fails, the server can switch over to the backup disk. Some

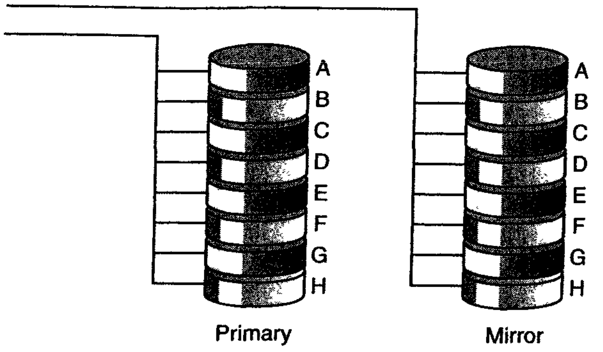


Figure 5–5 RAID 0+1 striping plus mirroring of data to two arrays

RAID techniques may be combined, as in level 0+1, to provide both striping throughput and data redundancy via mirroring, as shown in Figure 5–5.

The various techniques that RAID provides for redundancy and speed can be implemented by a dedicated RAID controller housed in the same enclosure as the disks, or by a RAID controller provisioned in the host system or file server. Data is passed from the operating system to the RAID controller, which then manages the striping or mirroring task. If the operating system itself manages data striping or mirroring, the host CPU must devote cycles to the task. The latter technique is called **software RAID**. In the next section, you will see how software RAID is implemented on JBODs.

Fibre Channel-attached RAID enclosures use an integrated RAID controller that sits behind the Fibre Channel interface. The RAID controller receives SCSI-3 requests via the loop or fabric and stores or retrieves data from the array using the appropriate RAID levels. From the standpoint of the Fibre Channel interconnect, the physical interface between the RAID controller and its disks is irrelevant. The RAID controller appears as an N_Port or NL_Port to the outside world but can use a proprietary bus, a SCSI bus, or other architecture to talk to its drives, as shown in Figure 5–6. For performance and reliability, however, RAID manufacturers can also incorporate Fibre Channel technology behind the RAID controller. This allows the use of Fibre Channel drives and the redundant topologies provided by, for example, dual-loop configurations. The use of arbitrated loop between the RAID controller and its Fibre Channel disks is invisible to the SAN, because the RAID controller still appears as a single N_Port or NL_Port to the rest of the topology.

Because a Fibre Channel RAID controller must support the functionality of an HBA and independent intelligence to manage striping and mirroring functions, it brings additional cost to a storage solution. Fibre Channel RAID

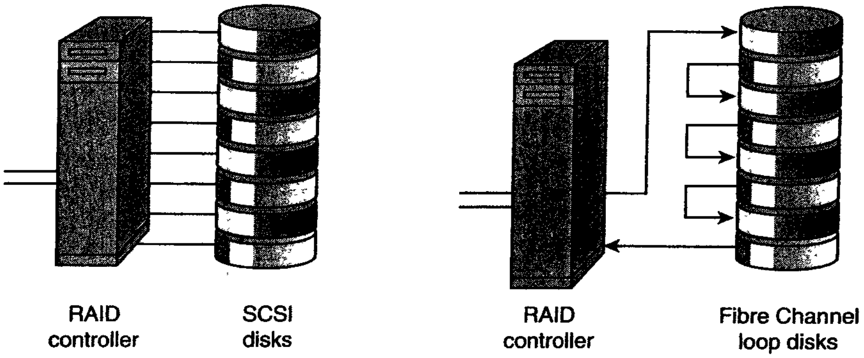


Figure 5–6 RAID with internal SCSI bus and internal arbitrated loop RAID

products justify the additional expense by including high-availability and storage management features required for enterprise networking. RAID enclosures offer redundant, hot-swappable power supplies, redundant fans, and self-diagnostics. Most of these products accommodate on-line spare drives for automatic reconstruction of data if a disk failure occurs. Large systems can be scaled to storage ranging from hundreds of gigabytes to tens of terabytes, and that implies an enterprise-level budget.

For mission-critical SANs, Fibre Channel RAIDs offer a number of advantages. The advanced diagnostics support, management features, reliability, and scalability typical of such high-end systems fill in most of the checkboxes for enterprise storage selection criteria. Server processing resources benefit by offloading data redundancy tasks to the RAID controller. Additionally, when the server is freed from the overhead of software RAID, it is also liberated from exclusive ownership of the storage device. No single server is responsible for managing the striping of data to multiple disks, so any server may own part or all of the RAID controller's data. Fibre Channel RAID thus enables server clustering and other applications that are predicated on co-ownership of storage resources.

From a SAN design standpoint, Fibre Channel RAIDs simplify configurations because the RAID controller appears as a single port address to the loop or fabric. Adding drives to a RAID enclosure does not add to the population of the SAN transport, even if the RAID enclosure internally uses arbitrated loop to access its disks. For arbitrated loops, you can attach more than a terabyte of storage without consuming all available AL_PAs. For direct fabric attachment, the RAID controller and all its storage appear as a single

N_Port on its own 100MBps segment, whereas a single JBOD enclosure will appear as multiple NL_Ports sharing a common bandwidth. However, JBODs do have one distinct advantage over RAIDs: cost.

5.4 Fibre Channel JBODs

A JBOD, or Just a Bunch of Disks, is an enclosure with multiple Fibre Channel disk drives inserted into a common backplane. The backplane provides the transmit-to-receive connections that bring the assembled drives into a common arbitrated loop segment; these connections bypass electronics that let you insert or remove drives without disrupting the loop circuit. Because Fibre Channel drives provide primary and secondary NL_Ports for dual-loop attachment, the JBOD enclosure typically includes external interfaces for connecting two loops, as shown in Figure 5–7.

A JBOD brings two things—the receive lead going to the first drive of a set, and the transmit lead coming from the last drive of a set—to a Fibre Channel interface, typically DB-9 copper. When the JBOD's interface is connected to a Fibre Channel hub or switch, the connection is not to a single Fibre Channel device but to multiple independent loop devices within the enclosure. An eight-drive JBOD, for example, appears as eight AL_PAs to the interconnection. If the connection is made to a switch, the switch port must be an FL_Port because the downstream enclosure is actually a loop segment. If the connection is to an arbitrated loop hub, the population of the entire loop is increased by the number of drives in the JBOD. JBODs, unlike Fibre Channel RAIDs, have a direct impact on the topology to which they are attached.

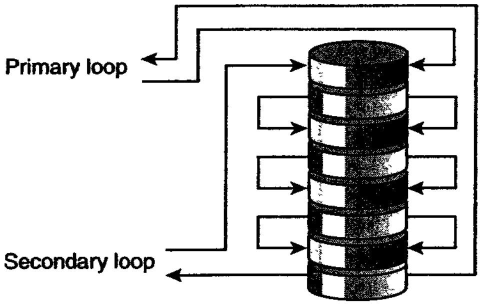


Figure 5–7 JBOD disk configuration with primary and secondary loop access

disk
disk drive
CONTEXT [Storage Device]
A non-volatile, randomly addressable, re-writable data storage device. This definition includes both rotating magnetic and optical disks and solid-state disks, or non-volatile electronic storage elements. It does not include specialized devices such as write-once-read-many (WORM) optical disks, nor does it include so-called RAM disks implemented using software to control a dedicated portion of a host computer's volatile random access memory.

disk array
CONTEXT [Storage System]
A set of disks from one or more commonly accessible disk subsystems, combined with a body of control software. The control software presents the disks' storage capacity to hosts as one or more virtual disks. Control software is often called firmware or microcode when it runs in a disk controller. Control software that runs in a host computer is usually called a volume manager.

disk array subsystem
CONTEXT [Storage System]
A disk subsystem which includes control software with the capability to organize its disks as disk arrays.

disk block
CONTEXT [Storage Device] [Storage System]
The unit in which data is stored and retrieved on a fixed block architecture disk. Disk blocks are of fixed usable size (with the most common being 512 bytes), and are usually numbered consecutively. Disk blocks are also the unit of on-disk protection against errors; whatever mechanism a disk employs to protect against data errors (e.g., ECC) protects individual blocks of data. cf. sector

disk cache
A cache that resides within a disk. A cache that resides in a controller or host whose primary purpose is to improve disk or array I/O performance. cf. cache, controller cache, host cache

disk image backup
CONTEXT [Backup] [Windows]
A backup consisting of a copy of each of the blocks comprising a disk's usable storage area. A disk image backup incorporates no information about

T10
CONTEXT [SCSI]
The ANSI T10 technical committee, the standards organization responsible for SCSI standards for communication between computers and storage subsystems and devices.

T11
CONTEXT [Fibre Channel]
The ANSI T11 technical committee, the standards organization responsible for Fibre Channel and certain other standards for moving electronic data into and out of computers and intelligent storage subsystems and devices.

tabular mapping
CONTEXT [Storage System]
A form of mapping in which a lookup table contains the correspondence between the two address spaces being mapped to each other. If a mapping between two address spaces is tabular, there is no mathematical formula that will convert addresses in one space to addresses in the other. cf. algorithmic mapping, dynamic mapping

tape
tape drive
tape transport
CONTEXT [Storage Device]
A storage device that writes data sequentially in the order in which it is delivered, and reads data in the order in which it is stored on the media. Unlike disks, tapes use implicit data addressing. cf. disk

tape array
CONTEXT [Storage System]
A collection of tapes from one or more commonly accessible storage subsystems, combined with a body of control software.

tape virtualization
tape drive virtualization
tape library virtualization
The act of creating abstracted tape devices by applying virtualization to tape drives or tape libraries.

target
CONTEXT [SCSI]
The system component that receives a SCSI I/O command. cf. initiator, LUN, target ID